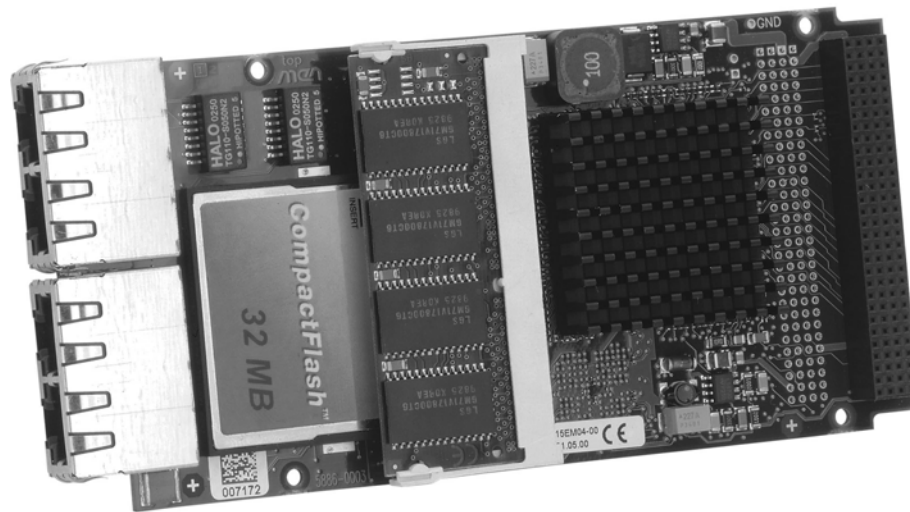


EM4N – ESM™ with PowerPC® MPC8245



User Manual

EM4N – ESM™ with PowerPC® MPC8245

The EM4N is a complete embedded single-board computer for use on any carrier board in different industrial environments. The final application consists either of a stand-alone EM4N, the EM4N with an application-specific carrier card and/or with additionally plugged PCI-104 modules.

The EM4N is controlled by an MPC8245 PowerPC® microprocessor which operates at 266 MHz up to 400 MHz. The MPC8245 includes a Floating Point Unit and a Memory Management Unit. The EM4N has an SO-DIMM socket for data and a CompactFlash® slot for program storage.

At its front panel the EM4N provides two RS232 interfaces and two Fast Ethernet channels. Additionally, nearly unlimited I/O functionality such as graphics controllers, serial interfaces, CAN bus controller etc. can be realized in a very flexible and cost-effective way by means of an on-board FPGA. Physical interfaces for such functions must be implemented on the carrier board. Compared to the EM4, the EM4N is equipped with a larger on-board FPGA, which has a capacity of 12,000 logic elements.

The EM4N comes with MENMON™ support. This firmware/BIOS can be used for bootstrapping operating systems (from disk, Flash or network), for hardware testing, or for debugging applications without running any operating system.

The EM4N is an ideal computer for low-cost deeply embedded solutions in very harsh environments, for machine control, Man-Machine Interfaces, fieldbus bridges or embedded Linux PCs.

For a first evaluation of the functions of the EM4N it is strongly recommended to use the EK2 ESM™ starter kit. The kit consists of the standard CPU module, an FPGA loaded with additional I/O functions, the carrier card with I/O connectors, an external PSU, VGA and RJ45 to D-Sub cables, and an adapter for mounting a PCI-104 module.

ESM™ modules are complete computers on a plug-on module. They consist of the hardware (CPU, chip set, memory, I/O) which is not fixed to any application-specific function, and an FPGA programmed in VHDL code, which provides I/O that is also still independent of a specific application. ESM™ modules are based on PCI. They have two system connectors: J1 has a fixed signal assignment, while J2 is variable depending on the final application-specific configuration of the ESM™ and the carrier board. J2 also feeds the I/O signals of the functions programmed in the FPGA to the carrier card.

Technical Data

CPU

- PowerPC®
 - MPC8245
 - 266MHz..400MHz

Memory

- L1 Cache integrated in MPC8245
- Up to 512MB SDRAM system memory
 - One 144-pin SO-DIMM slot for SDRAM modules
 - 133MHz memory bus frequency
- 2MB Flash
- Up to 16MB additional SDRAM, connected to FPGA, e.g. for video data
- Serial EEPROM 4kbit for factory settings
- CompactFlash® card interface for Flash ATA
 - Via on-board IDE
 - Type I
 - True IDE

Mass Storage

- Parallel IDE (PATA)
 - One port for hard-disk drives
 - Available via I/O connector
 - FPGA-controlled, see [FPGA Capabilities](#)
- CompactFlash® interface

Graphics

- Available via I/O connector
- FPGA-controlled, see [FPGA Capabilities](#)

I/O

- Ethernet
 - Two Ethernet 10/100Base-T channels
 - Two RJ45 connectors at front panel
 - Two on-board LEDs to signal LAN Link and Activity
- Two RS232 UARTs (COM1/COM2)
 - Two RJ45 connectors at front panel
 - Data rates up to 115kbits/s
 - 16-byte transmit/receive buffer
 - Handshake lines: none
- Further I/O possible through FPGA

Front Connections

- Two Ethernet (two RJ45 or one D-Sub)
- COM1/COM2 (two RJ45 or one D-Sub)

FPGA

- FPGA Altera® Cyclone™ EP1C12
 - 12,060 logic elements
 - 239,616 total RAM bits
 - Standard and user-defined functions
- Flexible configuration through IP cores
 - Available pin count: tbd.
- For available standard configurations and further options see [FPGA Capabilities](#).

PCI Interface

- 32-bit/33-MHz PCI interface at PCI-104 connector J1
- Support of two external masters

Miscellaneous

- Real-time clock
- Power supervision and watchdog

Electrical Specifications

- Supply voltage/power consumption:
 - +5V (-2%/+5%), 500mA
 - +3.3V (-2%/+5%), 1A, w/o SO-DIMM; increases up to 1.6A depending on installed SO-DIMM
- MTBF: 320,000h @ 40°C (derived from MIL-HDBK-217F)

Mechanical Specifications

- Dimensions: conforming to ESM™ specification (PCB: 149mm x 71mm), Type I-S
- Weight: 100g

Environmental Specifications

- Temperature range (operation):
 - 0..+60°C or -40..+85°C
 - Airflow: min. 10m³/h
- Temperature range (storage): -40..+85°C
- Relative humidity (operation): max. 95% non-condensing
- Relative humidity (storage): max. 95% non-condensing
- Altitude: -300m to + 3,000m
- Shock: 15g/11ms
- Bump: 10g/16ms
- Vibration (sinusoidal): 2g/10..150Hz
- Conformal coating on request

Safety

- PCB manufactured with a flammability rating of 94V-0 by UL recognized manufacturers

EMC

- Tested according to EN 55022 (radio disturbance), IEC1000-4-2 (ESD) and IEC1000-4-4 (burst) with regard to CE conformity

BIOS

- MENMON™

Software Support

- Linux (ELinOS)
- VxWorks®
- QNX® (on request)
- CANopen support: MEN Driver Interface System (MDIS™ for Windows®, Linux, VxWorks®, QNX®, RTX, OS-9®)
- MSCAN/Layer2 support: MEN Driver Interface System (MDIS™ for Windows®, Linux, VxWorks®, QNX®, RTX, OS-9®)

Non-Standard Options

Front Connections

- D-Sub connectors for Ethernet and COM

Mechanical

- PCI and I/O connectors can also be placed for face-to-face assembly (ESM™ Type N)

FPGA Capabilities

Functions through FPGA

- This MEN board offers the possibility to add customized I/O functionality in FPGA.
 - One on-board FPGA for multiple IP core functions
 - Functions available e.g. via I/O connector
- Number of functions and feasibility depends on pin counts and number of logic elements
 - FPGA capacity: 12,060 logic elements
 - Available pin count of on-board I/O connector: tbd.
 - Pin counts and logic elements of IP cores: see IP core compare chart

- Standard factory FPGA configuration:
 - Main bus interface
 - 16Z024_SRAM - SRAM controller (64KB)
 - 16Z023_IDE - IDE controller (PIO mode 0)
 - 16Z033_DISP - Display controller (800 x 600, 16-bit)
 - 16Z031_SPI - SPI touch panel controller
 - 16Z025_UART - UART controller (controls COM10..COM13)
 - 16Z029_CAN - CAN controller
 - 16Z022_GPIO - GPIO controller

MEN IP Cores

- MEN has a large number of standard IP cores to choose from.
- Examples:
 - IDE (e.g. PIO mode 0, UDMA mode 5)
 - UARTs
 - CAN bus
 - Display control
 - ...
- Depending on the hardware platform, SA-Adapters™ can be used to realize the physical lines.
- It depends on the board type which IP cores make sense and/or can be implemented. Please contact MEN for information on feasibility.
- MEN also offers development of new (customized) IP cores.

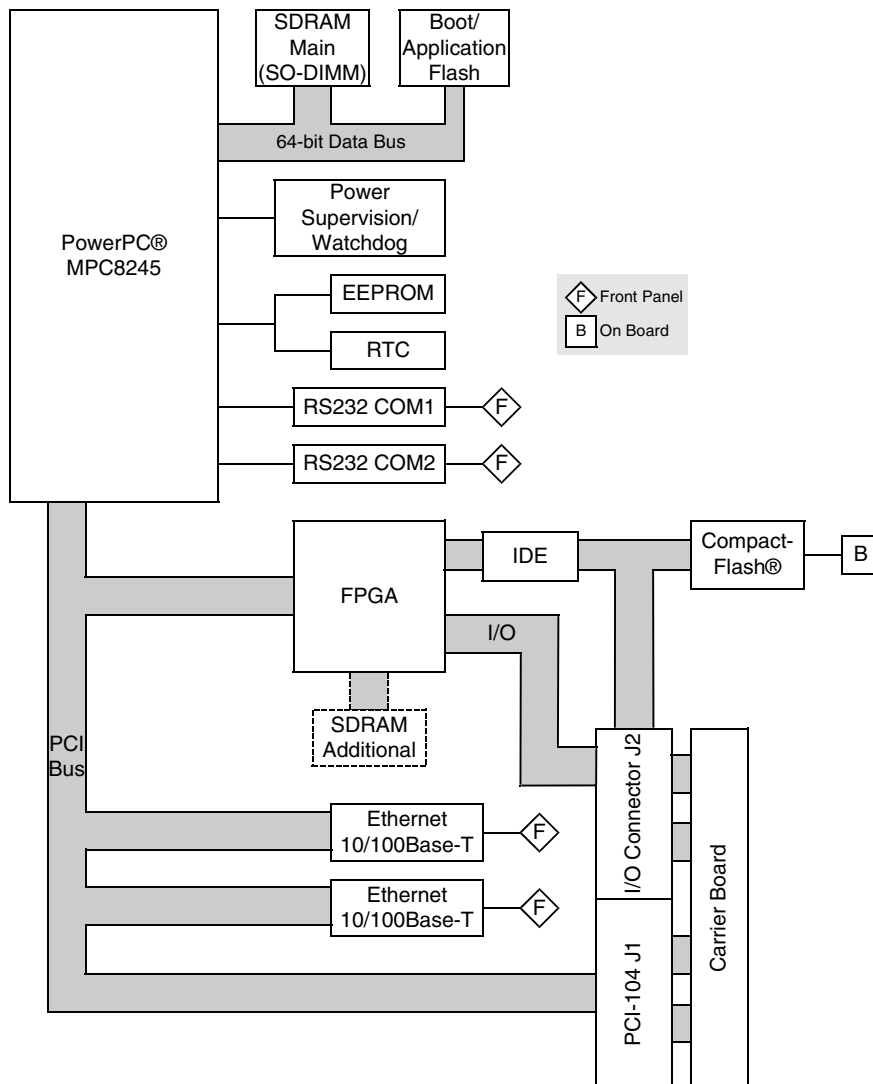
Third-Party IP Cores

- Third-party IP cores can also be used in combination with MEN IP cores.
- Examples:
 - www.altera.com
 - www.opencores.org

FPGA Design Software

- Altera® offers free download of Quartus® II Web Edition
 - Complete environment for FPGA and CPLD design
 - Includes schematic- and text-based design entry
 - Integrated VHDL and Verilog HDL synthesis and support for third-party synthesis software
 - SOPC Builder system generation software
 - Place-and-route, verification, and programming

Block Diagram



Product Safety



Electrostatic Discharge (ESD)

Computer boards and components contain electrostatic sensitive devices. Electrostatic discharge (ESD) can damage components. To protect the board and other components against damage from static electricity, you should follow some precautions whenever you work on your computer.

- Power down and unplug your computer system when working on the inside.
- Hold components by the edges and try not to touch the IC chips, leads, or circuitry.
- Use a grounded wrist strap before handling computer components.
- Place components on a grounded antistatic pad or on the bag that came with the component whenever the components are separated from the system.
- Store the board only in its original ESD-protected packaging. Retain the original packaging in case you need to return the board to MEN for repair.

About this Document

This user manual describes the hardware functions of the board, connection of peripheral devices and integration into a system. It also provides additional information for special applications and configurations of the board.

The manual does not include detailed information on individual components (data sheets etc.). A list of literature is given in the appendix.

History

Edition	Comments	Technical Content	Date of Issue
E1	First edition	H. Schubert, K. Popp, M. Beer	2006-01-19

Conventions



This sign marks important notes or warnings concerning proper functionality of the product described in this document. You should read them in any case.

italics

Folder and file names are printed in *italics*.

bold

Bold type is used for emphasis.

hyperlink

Hyperlinks are printed in [blue color](#).



The globe will show you where [hyperlinks](#) lead directly to the Internet, so you can look for the latest information online.

0xFF

Hexadecimal numbers are preceded by "0x", which is the usual C-language convention, and are printed in a monospace type, e.g. 0x00FFFF.

IRQ#
/IRQ

Signal names followed by "#" or preceded by a slash ("/") indicate that this signal is either active low or that it becomes active at a falling edge.

in/out

Signal directions in signal mnemonics tables generally refer to the corresponding board or component, "in" meaning "to the board or component", "out" meaning "coming from it".

Copyright Information

MEN Mikro Elektronik reserves the right to make changes without further notice to any products herein. MEN makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does MEN assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts.

MEN does not convey any license under its patent rights nor the rights of others.

Unless agreed otherwise, MEN products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the MEN product could create a situation where personal injury or death may occur. Should Buyer purchase or use MEN products for any such unintended or unauthorized application, Buyer shall indemnify and hold MEN and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that MEN was negligent regarding the design or manufacture of the part.

Unless agreed otherwise, the products of MEN Mikro Elektronik are not suited for use in nuclear reactors and for application in medical appliances used for therapeutical purposes. Application of MEN products in such plants is only possible after the user has precisely specified the operation environment and after MEN Mikro Elektronik has consequently adapted and released the product.

ESM™, MDIS4™, MENMON™, M-Module™, M-Modules™, SA-Adapter™, SA-Adapters™ and UBox™ are trademarks of MEN Mikro Elektronik GmbH. PC-MIP® is a registered trademark of MEN Micro, Inc. and SBS Technologies, Inc. MEN Mikro Elektronik® and the MEN logo are registered trademarks of MEN Mikro Elektronik GmbH.

Altera®, Avalon®, Cyclone™, Nios® and Quartus® are trademarks or registered trademarks of Altera Corp.

Freescape™ and PowerQUICC™ are trademarks of Freescape Semiconductor, Inc.

PowerPC® is a registered trademark of IBM Corp.

Microsoft® and Windows® are registered trademarks of Microsoft Corp.

OS-9®, OS-9000® and SoftStax® are registered trademarks of RadiSys Microware Communications Software Division, Inc. FasTrak™ and Hawk™ are trademarks of RadiSys Microware Communications Software Division, Inc. RadiSys® is a registered trademark of RadiSys Corporation.

QNX® is a registered trademark of QNX Ltd.

CompactFlash® is a registered trademark of SanDisk Corp.

Tornado® and VxWorks® are registered trademarks of Wind River Systems, Inc.

All other products or services mentioned in this publication are identified by the trademarks, service marks, or product names as designated by the companies who market those products. The trademarks and registered trademarks are held by the companies producing them. Inquiries concerning such trademarks should be made directly to those companies. All other brand or product names are trademarks or registered trademarks of their respective holders.

Information in this document has been carefully checked and is believed to be accurate as of the date of publication; however, no responsibility is assumed for inaccuracies. MEN Mikro Elektronik accepts no liability for consequential or incidental damages arising from the use of its products and reserves the right to make changes on the products herein without notice to improve reliability, function or design. MEN Mikro Elektronik does not assume any liability arising out of the application or use of the products described in this document.

Copyright © 2006 MEN Mikro Elektronik GmbH. All rights reserved.



Please recycle

Germany

MEN Mikro Elektronik GmbH
Neuwieder Straße 5-7
90411 Nuremberg
Phone +49-911-99 33 5-0
Fax +49-911-99 33 5-901
E-mail info@men.de
www.men.de

France

MEN Mikro Elektronik SA
18, rue René Cassin
ZA de la Châtelaine
74240 Gaillard
Phone +33 (0) 450-955-312
Fax +33 (0) 450-955-211
E-mail info@men-france.fr
www.men-france.fr

UK

MEN Micro Ltd
Whitehall, 75 School Lane
Hartford, Northwich
Cheshire UK, CW8 1PF
Phone +44 (0) 1606 781105
Fax +44 (0) 1606 784566
E-mail info@menmicro.co.uk
www.menmicro.co.uk

USA

MEN Micro, Inc.
PO Box 4160
Lago Vista, TX 78645-4160
Phone (512) 267-8883
Fax (512) 267-8803
E-mail sales@menmicro.com
www.menmicro.com

Contents

1	Getting Started	16
1.1	Map of the Board	16
1.2	Configuring the Hardware	16
1.3	Integrating the Board into a System	17
1.4	Installing Operating System Software	18
2	Functional Description	19
2.1	Power Supply	19
2.1.1	Watchdog	19
2.2	Clock Supply	19
2.3	Real-Time Clock	19
2.4	PowerPC CPU	20
2.4.1	General	20
2.4.2	Heat Sink	20
2.5	Bus Structure	21
2.5.1	Host-to-PCI Bridge	21
2.5.2	Local PCI Bus	21
2.6	Memory	22
2.6.1	DRAM System Memory	22
2.6.2	Flash	23
2.6.3	CompactFlash	23
2.6.4	Additional SDRAM	24
2.6.5	EEPROM	24
2.7	Ethernet Interface	25
2.7.1	Connection	25
2.7.2	General	27
2.7.3	10Base-T	27
2.7.4	100Base-T	27
2.8	UART COM1/COM2 Interfaces	28
2.9	I/O Connector J2	29
2.10	PCI-104 Interface J1	35
3	FPGA	37
3.1	General	37
3.2	System Unit	38
3.2.1	Functional Description	38
3.2.2	Address Organization	40
3.3	Standard Factory FPGA Configuration	45
3.3.1	IP Cores	45
3.3.2	FPGA Configuration Table	45

4	MENMON	46
4.1	General	46
4.1.1	State Diagram	47
4.2	Interacting with MENMON	49
4.2.1	Entering the Setup Menu/Command Line	49
4.3	Configuring MENMON for Automatic Boot	50
4.4	Calibrating the Touch Screen	50
4.5	Updating Boot Flash and CompactFlash	51
4.5.1	Update via the Serial Console using SERDL	51
4.5.2	Update from Network using NDL	51
4.5.3	Update via Program Update Menu	51
4.5.4	Automatic Update Check	52
4.6	Diagnostic Tests	53
4.6.1	Ethernet	53
4.6.2	SDRAM and SRAM	54
4.6.3	EEPROM	55
4.6.4	IDE/CompactFlash	56
4.6.5	COM1/COM2	57
4.6.6	Touch	57
4.6.7	Primary/Secondary MENMON	57
4.6.8	RTC	58
4.7	MENMON Configuration and Organization	59
4.7.1	Consoles	59
4.7.2	Video Modes	59
4.7.3	Abort Pin	59
4.7.4	MENMON Memory Map	60
4.7.5	MENMON BIOS Logical Units	61
4.7.6	System Parameters	62
4.8	MENMON Commands	67
5	Organization of the Board	69
5.1	Memory Mappings	69
5.2	Interrupt Handling	70
5.3	SMB Devices	70
5.4	PCI Devices on Bus 0	71
6	Appendix	72
6.1	Literature and Web Resources	72
6.1.1	PowerPC	72
6.1.2	EIDE	72
6.1.3	Ethernet	72
6.1.4	PCI-104	72

Figures

Figure 1. Map of the Board—Top View	16
Figure 2. SO-DIMM DRAM Installation	22
Figure 3. Installing a CompactFlash Card	23
Figure 4. FPGA — Block Diagram	37
Figure 5. MENMON — State Diagram, Degraded Mode/Full Mode	47
Figure 6. MENMON — State Diagram, Main State	48

Tables

Table 1.	Signal Mnemonics of Ethernet Connectors	25
Table 2.	Pin Assignment and Status LEDs of 8-pin RJ45 Ethernet Connectors	25
Table 3.	Pin Assignment of 9-pin D-Sub Ethernet Plug Connector (LAN1 and LAN2).	26
Table 4.	Signal Mnemonics of COM1/COM2 Connectors	28
Table 5.	Pin Assignment of 8-pin RJ45 COM1/COM2 Connectors	28
Table 6.	Pin Assignment of 9-pin D-Sub COM1/COM2 Plug Connector	28
Table 7.	Pin Assignment of I/O Connector J2 — General Pinout	30
Table 8.	Signal Mnemonics of I/O Connector J2 — General Pinout	31
Table 9.	Pin Assignment of I/O Connector J2 — Factory Standard FPGA Configuration	32
Table 10.	Signal Mnemonics of I/O Connector J2 — Factory Standard FPGA Configuration	33
Table 11.	Pin Assignment of PCI-104 J1	36
Table 12.	FPGA — Address Map.	40
Table 13.	FPGA — Factory Standard Configuration Table for EM4N	45
Table 14.	MENMON — Program Update Files and Locations	51
Table 15.	MENMON — Diagnostic Tests: Ethernet	53
Table 16.	MENMON — Diagnostic Tests: SDRAM and SRAM	54
Table 17.	MENMON — Diagnostic Tests: EEPROM.	55
Table 18.	MENMON — Diagnostic Tests: IDE/CompactFlash	56
Table 19.	MENMON — Diagnostic Tests: COM1/COM2	57
Table 20.	MENMON — Diagnostic Tests: Touch.	57
Table 21.	MENMON — Diagnostic Tests: Primary/Secondary MENMON.	57
Table 22.	MENMON — Diagnostic Tests: RTC	58
Table 23.	MENMON — System Parameters for Console Selection	59
Table 24.	MENMON — Address Map (Full-featured mode)	60
Table 25.	MENMON — Boot Flash Memory Map (2MB)	60
Table 26.	MENMON — Controller Logical Units (CLUNs).	61
Table 27.	MENMON — Device Logical Units (DLUNs)	61
Table 28.	MENMON — EM4N System Parameters — Autodetected Parameters	62
Table 29.	MENMON — EM4N System Parameters — Production Data	63
Table 30.	MENMON — EM4N System Parameters — Persistent Parameters	63
Table 31.	MENMON — EM4N System Parameters — VxWorks Bootline Parameters	65
Table 32.	MENMON — EM4N System Parameters — Carrier-Board Specific Parameters	65
Table 33.	MENMON — Reset Causes through System Parameter rststat.	66
Table 34.	MENMON — Command Reference	67
Table 35.	Memory Map — Processor View.	69
Table 36.	Address Mapping for PCI.	69

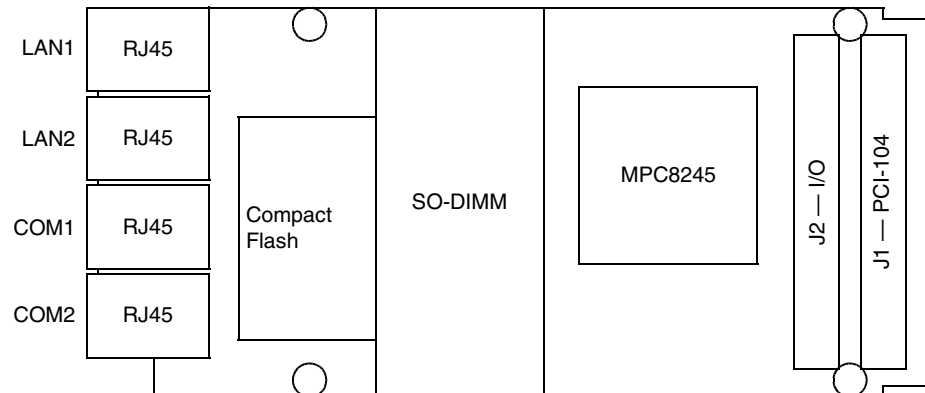
Table 37. BATS set up by MENMON	70
Table 38. SMB Devices	70
Table 39. PCI Devices on Bus 0	71

1 Getting Started

This chapter gives an overview of the board and some hints for first installation in a system.

1.1 Map of the Board

Figure 1. Map of the Board—Top View



1.2 Configuring the Hardware

You should check your hardware requirements before installing the board in a system, since most modifications are difficult or even impossible to do when the board is mounted on a carrier card.

The following check list gives an overview on what you might want to configure.

☒ **DRAM SO-DIMM modules**

The board may be shipped without any DRAM on board. You should check your main memory needs and install a suitable SO-DIMM module.

➤ Refer to [Chapter 2.6.1 DRAM System Memory on page 22](#) for a detailed installation description and hints on supported SO-DIMM modules.

☒ **CompactFlash**

The board is shipped without a CompactFlash card. You should check your needs and install a suitable CompactFlash card.

➤ Refer to [Chapter 2.6.3 CompactFlash on page 23](#) for a detailed installation description and hints on supported CompactFlash cards.

1.3 Integrating the Board into a System

You can use the following check list when installing the board in a system for the first time and with minimum configuration.



The board is completely trimmed on delivery.

- ☒ Power-down the system.
- ☒ Install the EM4N on your carrier card.
- ☒ Insert the assembly into your system.
- ☒ Connect a terminal to the standard RS232 interface COM1 (RJ45 connector).
- ☒ Set your terminal to the following protocol:
 - 9600 baud data transmission rate
 - 8 data bits
 - 1 stop bit
 - No parity
- ☒ Power-up the system.
- ☒ The terminal displays a message similar to the following:

```

Secondary MENMON for MEN EM04(N) 3.4
|-----|
| (c) 2002 - 2005 MEN Mikro Elektronik GmbH Nuremberg |
| MENMON 2nd Edition, Created Jun 24 2005      10:31:04 |
|-----|
| CPU Board: EM04N11          | CPU: MPC8245          |
| Serial Number: 12555        | CPU/MEM Clock: 384 / 128 MHz |
| HW Revision: 01.09.00       | DIMM Module: 64 MB 222      |
|-----|
| Carrier Board: EC04-00, Rev 00.00.00, Serial 20 |
| \-----/
Setting speed of NETIF 0 to AUTO

press 'ESC' for MENMON, 's' for setup
(Can't load BOOTLOGO.BMP)
Test SDRAM           : OK
Test ETHER0          : OK
Test SRAM             : OK
Test EEPROM          : OK
Test RTC              : OK
Test INT_CF           : OK
Test TOUCH            : OK
Test PRTCTRL          : OK

NOW AUTOEXECUTING: B0
No default start address configured. Stop.
Setup network interface CLUN 0x02, 00:c0:3a:21:2e:50 AUTO Searching for server (BOOTP)
in background Telnet daemon started on port 23 HTTP daemon started on port 80
MenMon>

```

- ☒ Now you can use the MENMON BIOS/firmware (see detailed description in [Chapter 4 MENMON on page 46](#)).
- ☒ Observe the installation instructions for the respective software.

1.4 Installing Operating System Software

The board supports Linux (ELinOS), VxWorks and QNX.



By standard, no operating system is installed on the board. Please refer to the operating system installation documentation on how to install the software!



You can find any software available on MEN's [website](#).

2 Functional Description

The following describes the individual functions of the board and their configuration on the board. There is no detailed description of the individual controller chips and the CPU. They can be obtained from the data sheets or data books of the semiconductor manufacturer concerned ([Chapter 6.1 Literature and Web Resources on page 72](#)).

2.1 Power Supply

The board is supplied with +5V and ± 3.3 V via PCI-104 connectors J1/J2.

The on-board power supply generates the 1.8V core voltage for the CPU and 1.5V core voltage for the FPGA.

2.1.1 Watchdog

A MAX6725 device supervises 5V, 2.5V and 1.8V and holds the CPU in reset condition until all supply voltages are within their nominal value range. In addition this device contains a watchdog that must be triggered. The watchdog timeout switches automatically from 54s (typ.) after reset to 1.68s (typ.) after the first trigger pulse. This allows a longer watchdog timeout period during the start-up phase.

After power-up the CPU loads the FPGA. The configuration file depends on the application. After configuration the FPGA serves the external hardware watchdog without further action by the CPU. If there is any problem loading the FPGA, the external watchdog causes a reset.

An additional watchdog is implemented in the FPGA.

2.2 Clock Supply

The clock supply generates all clocks for the on-board devices (PowerPC, SDRAM, PCI bus devices). The clock frequency is factory-set.

The local PCI bus operates at 32MHz.

2.3 Real-Time Clock

The board includes the 41T56 SMB real-time clock with integrated NVRAM. The real-time clock can be supplied via the J2 VBAT line. Interrupt generation of the RTC is not supported.

The 56-byte NVRAM is organized as a 56 bytes x 8 bits SRAM.

2.4 PowerPC CPU

The board is equipped with the MPC8245 processor, which includes a 32-bit superscalar PowerPC 603e core, the integrated host-to-PCI bridge, and two UARTs.

2.4.1 General

The PowerPC architecture is based on the POWER architecture implemented by the RS/6000™ family of computers. The PowerPC architecture takes advantage of recent technological advances in such areas as process technology, compiler design, and RISC microprocessor design to provide software compatibility across a diverse family of implementations, primarily single-chip microprocessors, intended for a wide range of systems.

2.4.2 Heat Sink

A heat sink may be provided to meet thermal requirements if a 400-MHz processor CPU is used.



Note: MEN gives no warranty on functionality and reliability of the ESM if you use any other heat sink than that supplied by MEN. Please contact either MEN directly or your local MEN sales office!

2.5 Bus Structure

2.5.1 Host-to-PCI Bridge

The integrated host-to-PCI bridge (internal in MPC8245) is used as host bridge and memory controller for the PowerPC processor. All transactions of the PowerPC to the PCI bus are controlled by the host bridge. The SDRAM and boot Flash are connected to the local memory bus of the integrated host-to-PCI bridge.

The PCI interface is PCI bus Rev. 2.2 compliant and supports all bus commands and transactions. Master and target operations are possible. Only big-endian operation is supported.

2.5.2 Local PCI Bus

The local PCI bus is controlled by the integrated host-to-PCI bridge. It runs at 32 MHz. The I/O voltage is fixed to 3.3V. The data width is 32 bits.

Major functional elements of the board, such as Ethernet, are connected to the local PCI bus.

2.6 Memory

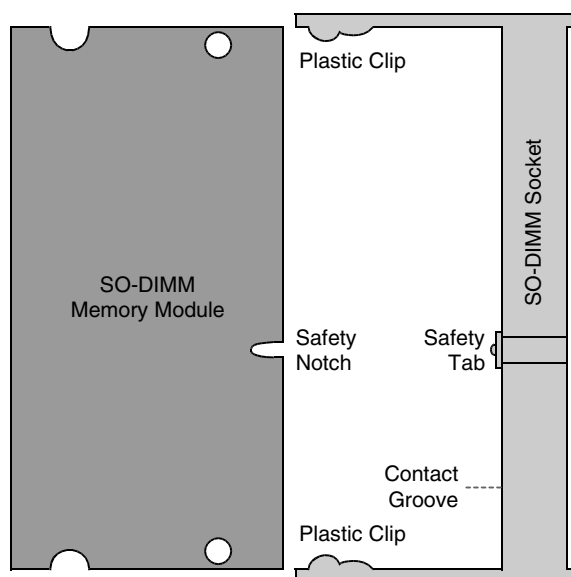
2.6.1 DRAM System Memory

One SDRAM bank (bank 0) is implemented on the board. Bank 0 is connected to a 144-pin SO-DIMM connector. The current board version supports SO-DIMM modules up to 512 MB.

2.6.1.1 Installing SO-DIMM DRAM

The board may be shipped without any DRAM SO-DIMM module installed. To install an SO-DIMM module, please stick to the following procedure.

Figure 2. SO-DIMM DRAM Installation



The DRAM module will only fit as shown above because of a safety tab on the SO-DIMM socket which requires a notch in the SO-DIMM module.



- ☑ Power down your system and remove the carrier card from the system.
- ☑ Deinstall the EM4N from the carrier card, if it is installed face-down.
- ☑ Place the memory module into the socket at a 45° angle and make sure that the safety tab and notch are aligned.
- ☑ Carefully push the memory module into the contact groove of the socket.
- ☑ Press the memory module down until it clicks into place.
- ☑ The plastic clips of the socket now hold the memory module in place.
- ☑ To release the module, squeeze both plastic clips outwards and carefully pull the module out of the socket.

2.6.1.2 Supported SO-DIMM Modules

You can install standard SO-DIMM modules with SDRAM components. See MEN's [website](#) for memory modules available from MEN.



Note: MEN gives no warranty on functionality and reliability of the board if you use any other module than that qualified and/or supplied by MEN. Please contact either MEN directly or your local MEN sales office.

2.6.2 Flash

The board has on-board Flash. It is controlled by the MPC8245 and can accommodate 2 MB. The data bus is 8 bits wide.

Flash memory contains the boot software for the MENMON/operating system bootstrapper and application software. The MENMON sectors are software-protected against illegal write transactions through a password in the serial download function of MENMON (cf. [Chapter 4.5 Updating Boot Flash and CompactFlash on page 51](#)).

2.6.3 CompactFlash

CompactFlash is a standard for small form factor ATA Flash drives. It is electrically compatible to the PC Card 1995 and PC Card ATA standards.

The CompactFlash standard is supported by industry's leading vendors of Flash cards and others.

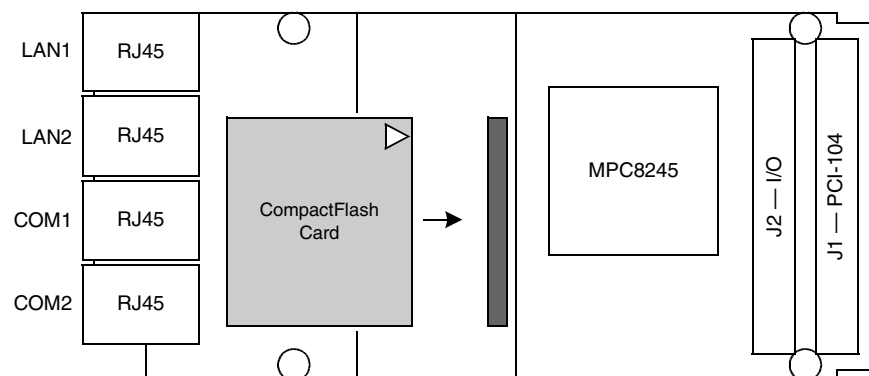
CompactFlash cards are operated in a True IDE Mode.

2.6.3.1 Installing a CompactFlash Card

The CompactFlash slot is within the SO-DIMM DRAM socket, i.e. the CompactFlash card is placed below a DRAM module.

The board is shipped without a CompactFlash card installed. To install CompactFlash, please stick to the following procedure.

Figure 3. Installing a CompactFlash Card



- ☑ Power down your system and remove the carrier card from the system.
- ☑ Deinstall the EM4N from the carrier card.
- ☑ If an SO-DIMM module is installed in the DRAM socket, remove the module as described in [Chapter 2.6.1.1 Installing SO-DIMM DRAM on page 22](#).
- ☑ Insert the CompactFlash card carefully as indicated by the arrow on top of the card, making sure that all the contacts are aligned properly and the card is firmly connected with the card connector.
- ☑ Reinstall your SO-DIMM module.
- ☑ To remove the CompactFlash card you must again remove and then reinstall the SO-DIMM module as described above.
- ☑ Observe manufacturer notes on usage of CompactFlash cards.

2.6.3.2 Supported CompactFlash Cards



The board supports standard CompactFlash cards. For CompactFlash cards available from MEN see MEN's [website](#).

2.6.4 Additional SDRAM

The board can be supplied with up to 16 MB additional SDRAM. It is controlled by the FPGA and can be used for graphics or other functions.

2.6.5 EEPROM

The board has a 4-kbit serial EEPROM for factory data, MENMON parameters, and for the VxWorks bootline.

2.7 Ethernet Interface

The EM4N has two Ethernet interfaces controlled by two Intel GD82551 devices. They support both 10Mbps/s and 100Mbps/s as well as full-duplex operation and autonegotiation.

Each of the two controllers has its own serial EPROM to store the MAC address etc.



Note: The unique Ethernet address is set at the factory and should not be changed. Any attempt to change this address may create node or bus contention and thereby render the board inoperable. A label on the Ethernet connectors of the board gives the set Ethernet address.

2.7.1 Connection

Two standard RJ45 connectors or one D-Sub connector are available at the front panel for connection to 10Base-T or 100Base-TX network environments. Two status LEDs for each connector are accommodated on the bottom side of the PCB, right next to the connectors, so as to be visible at the front.

The pin assignment corresponds to the Ethernet specification IEEE802.3.

Table 1. Signal Mnemonics of Ethernet Connectors

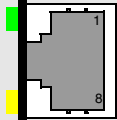
Signal	Direction	Function
Shield_R	-	Shield via RC network
RX+/-	in	Differential pair of receive data lines
TX+/-	out	Differential pair of transmit data lines

Connection via RJ45 Connectors

Connector types:

- Modular 8/8-pin mounting jack according to FCC68
- Mating connector:
Modular 8/8-pin plug according to FCC68

Table 2. Pin Assignment and Status LEDs of 8-pin RJ45 Ethernet Connectors

Lights up whenever there is transmit or receive activity	ACT 		1	TX+
			2	TX-
			3	RX+
			4	Shield_R
			5	Shield_R
			6	RX-
			7	Shield_R
			8	Shield_R
Lights up as soon as the link is established (10Base-T or 100Base-T)	LNK 			



Connection via 9-pin D-Sub Connector

A D-Sub connector can be implemented as an option. The two interfaces are routed to one D-Sub connector.

Connector types:

- 9-pin D-Sub plug according to DIN41652/MIL-C-24308, with thread bolt UNC 4-40
- Mating connector:
9-pin D-Sub receptacle according to DIN41652/MIL-C-24308, available for ribbon cable (insulation piercing connection), hand-soldering connection or crimp connection

Table 3. Pin Assignment of 9-pin D-Sub Ethernet Plug Connector (LAN1 and LAN2)

	1	LAN1_TX+
	2	LAN2_TX+
	3	-
	4	LAN2_RX+
	5	LAN1_RX+
	6	LAN1_TX-
	7	LAN2_TX-
	8	LAN2_RX-
	9	LAN1_RX-

2.7.2 General

Ethernet is a local-area network (LAN) protocol that uses a bus or star topology and supports data transfer rates of 100Mbps and more. The Ethernet specification served as the basis for the IEEE 802.3 standard, which specifies the physical and lower software layers. Ethernet uses the CSMA/CD access method to handle simultaneous demands. It is one of the most widely implemented LAN standards.

Ethernet networks provide high-speed data exchange in areas that require economical connection to a local communication medium carrying bursty traffic at high-peak data rates.

A classic Ethernet system consists of a backbone cable and connecting hardware (e.g. transceivers), which links the controllers of the individual stations via transceiver (transmitter-receiver) cables to this backbone cable and thus permits communication between the stations.

2.7.3 10Base-T

10Base-T is one of several adaptations of the Ethernet (IEEE 802.3) standard for Local Area Networks (LANs). The 10Base-T standard (also called Twisted Pair Ethernet) uses a twisted-pair cable with maximum lengths of 100 meters. The cable is thinner and more flexible than the coaxial cable used for the 10Base-2 or 10Base-5 standards. Since it is also cheaper, it is the preferable solution for cost-sensitive applications.

Cables in the 10Base-T system connect with RJ45 connectors. A star topology is common with 12 or more computers connected directly to a hub or concentrator.

The 10Base-T system operates at 10Mbps and uses baseband transmission methods.

2.7.4 100Base-T

The 100Base-T networking standard supports data transfer rates up to 100Mbps. 100Base-T is actually based on the older Ethernet standard. Because it is 10 times faster than Ethernet, it is often referred to as Fast Ethernet. Officially, the 100Base-T standard is IEEE 802.3u.

Like Ethernet, 100Base-T is based on the CSMA/CD LAN access method. There are several different cabling schemes that can be used with 100Base-T, e.g. 100Base-TX, with two pairs of high-quality twisted-pair wires.

2.8 UART COM1/COM2 Interfaces

COM1/COM2 are standard RS232 interfaces led to two RJ45 connectors or one D-Sub connector at the front panel. Both serial interfaces are controlled by the MPC8245.

Table 4. Signal Mnemonics of COM1/COM2 Connectors

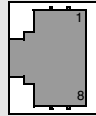
Signal	Direction	Function
GND	-	Ground
RXD	in	Receive data
TXD	out	Transmit data

Connection via RJ45 Connectors

Connector types:

- Modular 8/8-pin mounting jack according to FCC68
- Mating connector:
Modular 8/8-pin plug according to FCC68

Table 5. Pin Assignment of 8-pin RJ45 COM1/COM2 Connectors

	1	-
	2	-
	3	-
	4	GND
	5	RXD
	6	TXD
	7	-
	8	-

Connection via 9-pin D-Sub Connector

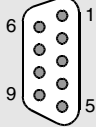


A D-Sub connector can be implemented as an option. The two interfaces are routed to one D-Sub connector.

Connector types:

- 9-pin D-Sub plug according to DIN41652/MIL-C-24308, with thread bolt UNC 4-40
- Mating connector:
9-pin D-Sub receptacle according to DIN41652/MIL-C-24308, available for ribbon cable (insulation piercing connection), hand-soldering connection or crimp connection

Table 6. Pin Assignment of 9-pin D-Sub COM1/COM2 Plug Connector

	1	COM1_GND
	2	COM2_RXD
	3	COM2_TXD
	4	COM1_TXD
	5	COM2_GND
6	-	
7	-	
8	-	
9	COM1_RXD	

2.9 I/O Connector J2

The board features a second 120-pin PCI-104-standard connector that implements additional I/O. The type of I/O depends on the FPGA configuration of the EM4N, which is very flexible and can contain a number of FPGA IP cores. For more information, please refer to [Chapter 3 FPGA on page 37](#).

To illustrate the possibilities, this manual gives the standard factory FPGA configuration that is used on model 15EM04N00.

This version provides the following interfaces via J2:

(See [Chapter 3.3 Standard Factory FPGA Configuration on page 45](#).)

- IDE
- SRAM
- Display
- SPI touch panel
- UARTs COM10..COM13
- CAN bus
- GPIO
- Miscellaneous functions

The following tables give the pinouts of the raw J2 connector (FPGA-independent) and of its signals in conjunction with the above-mentioned FPGA configuration.

Connector types:

- 4-row, 120-pin PCI-104 receptacle connector, 2mm pitch
- Mating connector:
4-row, 120-pin PCI-104 plug connector, 2mm pitch

Table 7. Pin Assignment of I/O Connector J2 — General Pinout

		A	B	C	D
	1	PA1	PB1	+5V	PD1
	2	GND	PB2	PC2	+5V
	3	PA3	GND	PC3	PD3
	4	PA4	PB4	GND	PD4
	5	+3.3V	PB5	PC5	GND
	6	PA6	+3.3V	PC6	PD6
	7	PA7	PB7	+3.3V	PD7
	8	GND	PB8	PC8	+3.3V
	9	PA9	GND	PC9	PD9
	10	PA10	PB10	GND	PD10
	11	+5V	PB11	PC11	GND
	12	PA12	+5V	PC12	PD12
	13	PA13	PB13	+5V	PD13
	14	GND	PB14	PC14	+5V
	15	PA15	GND	PC15	PD15
	16	PA16	PB16	GND	PD16
	17	+3.3V	PB17	PC17	GND
	18	PA18	+3.3V	PC18	PD18
	19	PA19	PB19	+3.3V	PD19
	20	GND	PB20	PC20	+3.3V
	21	PA21	GND	PC21	PD21
	22	PA22	PB22	GND	PD22
	23	+5V	PB23	PC23	GND
	24	PA24	+5V	PC24	PD24
	25	PA25	PB25	+5V	PD25
	26	GND	PB26	PC26	+5V
	27	PA27	GND	PC27	PD27
	28	PA28	PB28	GND	PD28
	29	+5V	SDA	PC29	GND
	30	SCL	+3.3V_STBY	PC30	PD30

Diagram of the 30-pin connector J2 showing the pin layout. The pins are arranged in a 4x8 grid with columns labeled A, B, C, and D. The rows are numbered 1 to 30. The diagram shows the physical arrangement of the pins, with some pins highlighted in green and others in red to match the table.

Table 8. Signal Mnemonics of I/O Connector J2 — General Pinout

	Signal	Direction	Function
Power	+3.3V	-	+3.3V power supply
	+3.3V_STBY	in	Power supply for real-time clock
	+5V	-	+5V power supply
	GND	-	Digital ground
FPGA I/O	PAxx..PDxx	in/out	FPGA general-purpose I/O lines
I²C EEPROM	SCL	in/out	I ² C bus
	SDA	in/out	I ² C bus

Table 9. Pin Assignment of I/O Connector J2 — Factory Standard FPGA Configuration

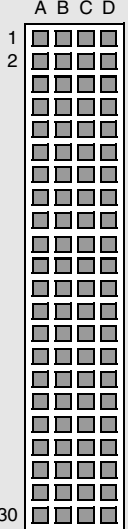
		A	B	C	D
	1	IDE_RST#	SCLK	+5V	BLUE[2]
	2	GND	SDI	RXD10	+5V
	3	IDE_D7	GND	RTS10#	BLUE[3]
	4	IDE_D6	IDE_D8	GND	BLUE[4]
	5	+3.3V	IDE_D9	CTS10#	GND
	6	IDE_D5	+3.3V	TXD12	BLUE[5]
	7	IDE_D4	IDE_D10	+3.3V	GREEN[0]
	8	GND	IDE_D11	RXD12	+3.3V
	9	IDE_D3	GND	RTS12#	GREEN[1]
	10	IDE_D2	IDE_D12	GND	GREEN[2]
	11	+5V	IDE_D13	CTS12#	GND
	12	IDE_D1	+5V	TXD11	GREEN[3]
	13	IDE_D0	IDE_D14	+5V	GREEN[4]
	14	GND	IDE_D15	RXD11	+5V
	15	Reserved	GND	Reserved	GREEN[5]
	16	IDE_WR#	SDO	GND	RED[0]
	17	+3.3V	SCS#	Reserved	GND
	18	IDE_RD#	+3.3V	Reserved	RED[1]
	19	Reserved	IDE_EN#	+3.3V	RED[2]
	20	GND	PENIRQ#	COM10_SW	+3.3V
	21	Reserved	GND	Reserved	RED[3]
	22	IDE_IRQ	Reserved	GND	RED[4]
	23	+5V	PWR_FAIL	PBRST#	GND
	24	IDE_A1	+5V	CAN_TXD	RED[5]
	25	IDE_A0	TXD10	+5V	DOTCLK
	26	GND	IDE_A2	RXD13	+5V
	27	IDE_CS1#	GND	TXD13	DTMG
	28	CAN_RXD	IDE_CS3#	GND	HSYNC
	29	+5V	SDA	BLUE[0]	GND
	30	SCL	+3.3V_STBY	BLUE[1]	VSYNC

Table 10. Signal Mnemonics of I/O Connector J2 — Factory Standard FPGA Configuration

	Signal	Direction	Function
Power	+3.3V	-	+3.3V power supply
	+3.3V_STBY	in	Power supply for real-time clock
	+5V	-	+5V power supply
	GND	-	Digital ground of respective interface
I²C EEPROM	SCL	out	I ² C bus
	SDA	in/out	I ² C bus
IDE/SRAM	IDE_A[2:0]	out	IDE address [2:0]
	IDE_CS1#	out	IDE chip select 1
	IDE_CS3#	out	IDE chip select 3
	IDE_D[15:0]	in/out	IDE data [15:0]
	IDE_EN#	out	IDE enable (0 = IDE)
	IDE_IRQ	in	IDE interrupt request
	IDE_RD#	out	IDE read strobe
	IDE_RST#	out	IDE reset
	IDE_WR#	out	IDE write strobe
UARTs	CTS10#	in	COM10 clear to send
	COM10_SW	out	COM10 mode 0 = COM10 operates in RS422/485 mode 1 = COM10 operates in RS232 mode
	RTS10#	out	COM10 request to send
	RXD10	in	COM10 receive data
	TXD10	out	COM10 transmit data
	RXD11	in	COM11 receive data
	TXD11	out	COM11 transmit data
	CTS12#	in	COM12 clear to send
	RTS12#	out	COM12 request to send
	RXD12	in	COM12 receive data
	TXD12	out	COM12 transmit data
	RXD13	in	COM13 receive data
	TXD13	out	COM13 transmit data
CAN Bus	CAN_RXD	in	CAN bus receive data
	CAN_TXD	out	CAN bus transmit data

	Signal	Direction	Function
Display	DOTCLK	out	Dot clock
	DTMG	out	Display data valid/invalid
	HSYNC	out	Horizontal synchronization
	RED[5:0], GREEN[5:0], BLUE[5:0]	out	Monitor interface (red, green, blue)
	VSYNC	out	Vertical synchronization
SPI Touch Control	PENIRQ#	in	Touch controller interrupt
	SCLK	out	SPI clock
	SDI	in	SPI data in
	SDO	out	SPI data out
	SCS#	out	SPI chip select
Other	PBRST#	in	Push button reset
	PWR_FAIL	in	Power supply fail

2.10 PCI-104 Interface J1

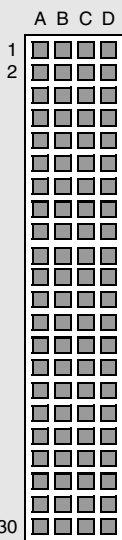
The EM4N provides a 32-bit PCI interface at the PCI-104 connector J1. The EM4N is always the system controller of the PCI-104 bus and supports two external masters.

Connector types:

- 4-row, 120-pin PCI-104 receptacle connector, 2mm pitch, e.g. Samtec ESQT-130-02-G-Q-368
- Mating connector:
4-row, 120-pin PCI-104 plug connector, 2mm pitch

Table 11. Pin Assignment of PCI-104 J1

		A	B	C	D
	1	64EN#	Reserved	+5V	AD00
	2	VI/O +3.3V	AD02	AD01	+5V
	3	AD05	GND	AD04	AD03
	4	C/BE0#	AD07	GND	AD06
	5	GND	AD09	AD08	GND
	6	AD11	VI/O +3.3V	AD10	M66EN
	7	AD14	AD13	GND	AD12
	8	+3.3V	C/BE1#	AD15	+3.3V
	9	SERR#	GND	SB0#	PAR
	10	GND	PERR#	+3.3V	SDONE
	11	STOP#	+3.3V	LOCK#	GND
	12	+3.3V	TRDY#	GND	DEVSEL#
	13	FRAME#	GND	IRDY#	+3.3V
	14	GND	AD16	+3.3V	C/BE2#
	15	AD18	+3.3V	AD17	GND
	16	AD21	AD20	GND	AD19
	17	+3.3V	AD23	AD22	+3.3V
	18	IDSEL0	GND	IDSEL1	IDSEL2
	19	AD24	C/BE3#	VI/O +3.3V	IDSEL3
	20	GND	AD26	AD25	GND
	21	AD29	+5V	AD28	AD27
	22	+5V	AD30	GND	AD31
	23	REQ0#	GND	REQ1#	VI/O +3.3V
	24	GND	REQ2#	+5V	GNT0#
	25	GNT1#	VI/O +3.3V	GNT2#	GND
	26	+5V	CLK0	GND	CLK1
	27	CLK2	+5V	CLK3	GND
	28	GND	INTD#	+5V	RST#
	29	+12V	INTA#	INTB#	INTC#
	30	-12V	REQ3#	GNT3#	REQ64#



For a description of signals please refer to the PCI-104 specification see [Chapter 6.1 Literature and Web Resources](#) on page 72.

3 FPGA

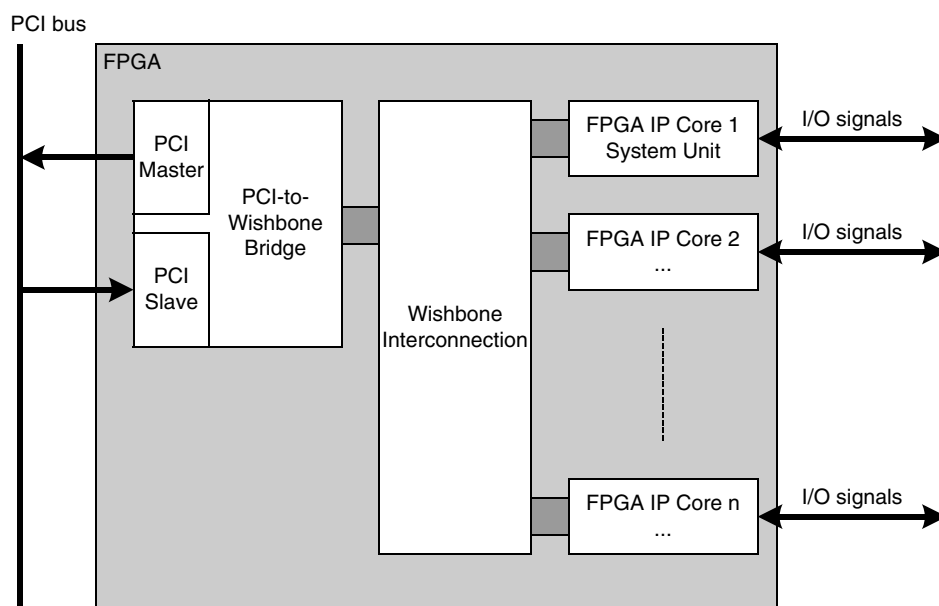
3.1 General

The FPGA—as a part of the EM4N—represents an interface between a user-selectable configuration of I/O modules (IP cores) and the PCI bus. The PCI core included in the FPGA can be a PCI target or master. It can be accessed via memory single/burst read/write cycles.

The Wishbone bus is the uniform interface where IP cores can be connected in addition to the System Unit to provide the highest possible flexibility for different configurations of the FPGA.

Each implementation contains a bridge from the PCI bus to the Wishbone bus. Additionally each implementation contains a system unit for system-specific functions such as reset/interrupt control or watchdog etc. and the system library. The presence of the single system unit functions (and system registers) depends on the necessity in the actual implementation and cannot be described in general.

Figure 4. FPGA — Block Diagram



The **FPGA System Unit** contains a configuration table providing the information which modules are implemented (device number) in the current configuration. Furthermore the revision, the instance number (one module can be instantiated more than one time), the interrupt routing and the base address of the module are stored. At initialization time, the CPU has to read the configuration table to get the information of the base addresses of the included modules.

Note that with regard to the FPGA resources such as available logic elements or pins it is not possible to grant all possible combinations of the FPGA IP cores. [Chapter 3.3 Standard Factory FPGA Configuration on page 45](#) describes one possible configuration of the FPGA. Please ask our [sales staff](#) for other configurations.



3.2 System Unit

The system unit is available in all configurations and contains a number of modules with system functionality communicating with the CPU.

3.2.1 Functional Description

3.2.1.1 Interrupt Controller

The interrupt controller combines the interrupt requests of all implemented modules, one interrupt line for one module. The system registers are consolidated to one module, therefore interrupt line 0 is reserved for the system register unit. Since the interrupt controller handles 16 interrupt requests, 15 IP cores can request an interrupt without sharing.

The PowerPC microcontroller which is used to interact with the FPGA contains an interrupt controller, so it is not necessary to prioritize the interrupt requests. This is done by the microcontroller. The interrupt requests are stored in a 16-bit [Interrupt Request Register \(IRQR\)](#) and are periodically sent to the interrupt controller of the PowerPC through serial I/O lines. Any module's interrupt request can be enabled or disabled only in the module which generates the interrupt by writing 1 or 0 to the corresponding bit of the module's interrupt enable register.

The CPU has to reset an interrupt request in the module which generates the interrupt.

3.2.1.2 Reset Controller

Since there are several reset possibilities the CPU can read the information which reset condition occurred the last time. Therefore, the board's FPGA contains a reset controller with a reset register which stores the cause of the occurred restart.

A system reset can occur through:

- Reset button
- Internal watchdog timer expiration
- External *HRESET* (e.g. by external watchdog timer expiration)
- Software reset

The reset controller notices the reset cause by watching its reset inputs and stores the cause in the [Reset Cause Register \(RCR\)](#). This register must not be set back by the system reset, so that the CPU is able to read the reset cause after system restart from the board's FPGA Reset Cause Register. The CPU can reset the bits in the reset cause register by writing 1 to the corresponding bit.

The reset controller is able to generate an interrupt on a power supply fail indication. The power fail detection has to be done on the ESM carrier board. The time between the indication and the failure of 5V/3.3V can be used by the CPU to save the date and time of exception. This interrupt can be enabled or disabled by handling the [RC Control Register \(RCCR\)](#).

The bidirectional power fail port can be used for two different functions. If it is used as an output, a heartbeat LED can be driven with the value of the control register bit *HB*. If it is used as an input, the port checks the level of the power fail signal to

detect a loss of the power supply. To switch between input and output, a 16-bit counter controls an output-enable signal. The counter value 0x1 switches the port to an input, all other values force the port to act as an output.

The reset controller contains a system watchdog which has to be triggered by the CPU through alternating write operations (0xAA and 0x55) in configurable time intervals. If the time interval exceeds without a correct write operation, the watchdog executes a system reset by forcing *HRSTN* to 0. Additionally bit *WDEX* is set in the [Reset Cause Register](#) to provide the reset cause information after system restart.

3.2.2 Address Organization

3.2.2.1 Address Map

Table 12. FPGA — Address Map

Address	D15..D0
0x0000	Identification Word (IW) (r)
0x0004	Magic Word (MW) (r)
0x0008	Configuration Table (CT) (r)
0x0080	Interrupt Request Register (IRQR) (r)
0x0088	Reset Cause Register (RCR) (r)
0x008C	Watchdog Timer Register (WDTR) (r/w)
0x0090	Watchdog Value Register (WDVR) (r/w)
0x0094	Reset Controller Control Register (RCCR) (r/w)
0x0098	Reset Controller Interrupt Request Register (RCIRQR) (r/w)
0x009C..0x00FF	Reserved
0x0100..0x1FFF	FPGA IP cores (see detailed address map in the respective IP core user manual)

3.2.2.2 Identification Word and Configuration Table

The address space of the FPGA starts with an identification word and a configuration table. The identification word (BAR0; address 0x0000) describes the FPGA configuration by a character (A to Z) and its revision number. Byte 0 at address 0x0000 contains the revision number, byte 1 at address 0x0001 contains the describing character in ASCII format.

Example: address 0x0000 $\Rightarrow$ identification word 0x4103 $\Rightarrow$ variant A3

In the FPGA address space the identification word is followed by the configuration table which gives detailed information about the implemented modules.

The Magic Word at address 0x0004 is used by software to identify the configuration table.

It is possible to use different Base Address Registers of the PCI Configuration Space. (This is necessary for modules which require an address space larger than 256 bytes.) If other BARs are used, the module's memory space which is allocated with the BARs other than BAR 0 has other base addresses as viewed from the PCI bus. Inside the board's FPGA the module's address space starts at 0x0000 too, but all modules are enabled by a chip select signal generated in consideration of the used BAR.

A separate Base Address Register (apart from BAR 0) can be used e.g. to access memory. This simplifies PCI host accesses to that module because the memory's address always starts at 0x0000. The PCI host must access the FPGA with the correct base address only.

To provide the information which modules are implemented, the FPGA configuration table contains two binary words for each module to code module information such as

- Device
- Revision
- Used Base Address Register
- Number of instance
- Used interrupt line for that device
- Start address in EM4N memory space



You can find an overview of all available FPGA IP cores on MEN's [website](#). You can find a detailed description of each FPGA IP core in the respective user manual also available on MEN's website.

[Chapter 3.3 Standard Factory FPGA Configuration on page 45](#) gives an example configuration, including a configuration table.

3.2.2.3 Registers

Interrupt Request Register IRQR (0x0080) (read only)

The read-only Interrupt Request Register IRQR provides information about the interrupt source that has generated an interrupt. Each implemented module supplies one Interrupt Request Register bit. The configuration table gives the information about the interrupt routing to show which module corresponds to which IRQR bit.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Irqn S15	Irqn S14	Irqn S13	Irqn S12	Irqn S11	Irqn S10	Irqn S9	Irqn S8	Irqn S7	Irqn S6	Irqn S5	Irqn S4	Irqn S3	Irqn S2	Irqn S1	Irqn S0

Reset value: 0xFFFF

IrqnSx 0 = Pending interrupt of source (module) *x*
 1 = No interrupt of source (module) *x* is pending

Reset Cause Register RCR (0x0088) (read/write)

The RCR provides information which reset source has activated a restart. The register will not be reset by any reset source, so the CPU can still read the cause of the reset after the accomplished restart. To reset a bit of the RCR, the CPU has to write 1 (acknowledge) to the corresponding bit.

15..4	3	2	1	0
Reserved	SW_RST	RSTBUT	WDEX	HRSTN

Reset value: Module will not be reset by any reset, only by configuration of FPGA (Config. Value: 0x0000)

SW_RST 0 = No software reset occurred
 1 = Software reset occurred by setting bit 1 of RCCR

RSTBUT 0 = External reset button not activated
 1 = External reset button was activated

WDEX 0 = Internal watchdog triggered correctly
 1 = Internal watchdog expired (not triggered by CPU)

HRSTN 0 = No *HRESET* occurred
 1 = *HRESET* from an external module occurred

Watchdog Timer Register WDTR (0x008C) (read/write)

The WDTR contains both the watchdog expiration time and the watchdog enable function. Bit 15 enables or disables the internal watchdog while bits 9..0 are used to load a system clock relative clocked counter. If the down-counter reaches zero before the CPU has triggered the watchdog, a watchdog expiration occurs.

15	14..10	9..0
WDEN	Reserved	Watchdog expiration time

Reset value: 0x0000

WDEN 0 = Internal watchdog disabled

 1 = Internal watchdog enabled

Relation between system clock and watchdog clock:

$$T_{count_clock} = 2^{16} \cdot T_{sys_clock}$$

Example:

$$f_{sys_clock} = 32 \text{ MHz}; \quad T_{sys_clock} = 31.25 \text{ ns}$$

$$\Rightarrow T_{count_clock} = 65536 \cdot 31.25 \text{ ns} = 2.048 \text{ ms}$$

The watchdog expiration time can be calculated by the following equation:

$$Wd_{extime} = T_{count_clock} \cdot Wd_{expiration\ time} = 2.048 \text{ ms} \cdot Wd_{expiration\ time}$$

Example:

$$\text{Watchdog expiration time} = 0000111111\text{bin}$$

$$Wd_{expiration\ time} = T_{count_clock} \cdot 63 \text{ dez} = 2.048 \text{ ms} \cdot 63 = 0.129 \text{ s}$$

This calculation leads to a user-configurable watchdog expiration time interval of:

$$\text{Register Value} = 0000000000\text{bin} \Rightarrow Wd_{extime} = 2.048 \text{ ms}$$

$$\text{Register Value} = 1111111111\text{bin} \Rightarrow Wd_{extime} = 2.095 \text{ s}$$

Watchdog Value Register WDVR (0x0090) (read/write)

The WDVR contains the alternating watchdog trigger value. The watchdog has to be triggered by the CPU with write operations using the alternating values 0xAAAA and 0x5555.

15..0
Watchdog trigger value

Reset value: 0x0000

Reset Controller Control Register RCCR (0x0094) (read/write)

The RCCR enables or disables the interrupt request from the reset controller by setting bit *RCIREN* to 1 or 0 and offers the possibility to reset the EM4N through a CPU register access (bit *SW_RST*). Writing 1 to bit *SW_RST* forces a reset of the complete module including the FPGA. Bit *HB* is used to trigger a heartbeat LED.

15..3	2	1	0
Reserved for future use	HB	SW_RST	RCIREN

Reset value: 0x0000

HB 0 = Heartbeat LED turned on
 1 = Heartbeat LED turned off

SW_RST 0 = CPU reset inactive
 1 = CPU reset active

RCIREN 0 = Reset controller interrupt request disabled
 1 = Reset controller interrupt request enabled

Reset Controller Interrupt Request Register RCIRQR (0x0098) (read/write)

The RCIRQR (only bit 0 is used) requests a pending reset controller interrupt when bit *RCIRQ* is set to 1. Writing 1 to bit *RCIRQ* resets the interrupt request.

15..1	0
Reserved for future use	RCIRQ

Reset value: 0x0000

RCIRQ 0 = No reset controller interrupt request
 1 = Reset controller interrupt request pending

3.3 Standard Factory FPGA Configuration

3.3.1 IP Cores

The factory FPGA configuration for standard boards comprises the following FPGA IP cores:

- 16Z035_SYSTEM — System Unit
- 16Z024_SRAM — SRAM controller
- 16Z023_IDE — IDE controller
- 16Z033_DISP — Display controller
- 16Z031_SPI — SPI touch panel controller
- 16Z025_UART — UART controller (controls COM10..13)
- 16Z029_CAN — CAN controller
- 16Z022_GPIO — GPIO controller

The configuration matches the pin assignment given in this manual for the J2 I/O connector.

3.3.2 FPGA Configuration Table

The resulting configuration table of the standard FPGA configuration is as follows:

Table 13. FPGA — Factory Standard Configuration Table for EM4N

IP Core	Device ID	Revision	IRQ	BAR	Offset
SysMod	0	7	0	0	0
IDE Controller	4	6	1	0	100
SPI	3	4	2	0	200
UART	7	C	3	0	300
CAN	8	5	4	0	400
GPIO	9	3	5	0	500
SRAM	6	6	6	1	0
Display	1	7	7	2	0
User module	3E	0	8	3	0
INTA	B	1	9	0	0
INTB	B	1	A	0	0
INTC	B	1	B	0	0
INTD	B	1	C	0	0
Magic Word	ABCD		All values are given in hexadecimal notation.		
Variant	F				
Revision	7				

4 MENMON

4.1 General

MENMON is the CPU board firmware that is invoked when the system is powered on.

The basic tasks of MENMON are:

- Initialize the CPU and its peripherals.
- Load the FPGA code (if applicable).
- PCI auto configuration.
- Perform self-test.
- Provide debug/diagnostic features on MENMON command line.
- Interaction with the user via touch panel/TFT display (if supported by FPGA).
- Boot operating system.
- Update firmware or operating system.



The following description only includes board-specific features. For a general description and in-depth details on MENMON 3.x, please refer to the [MENMON 2nd Edition User Manual](#).

4.1.1 State Diagram

Figure 5. MENMON — State Diagram, Degraded Mode/Full Mode

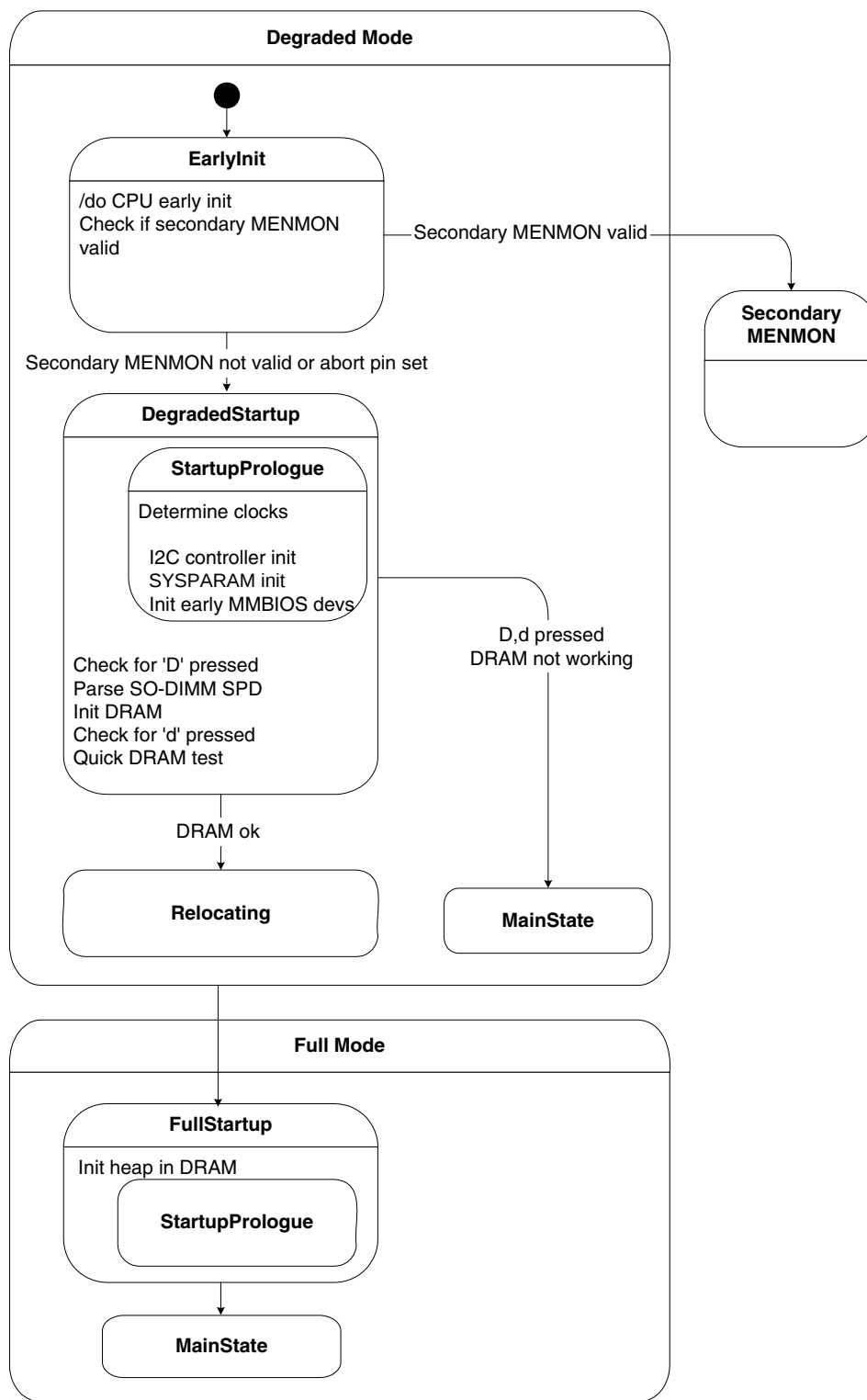
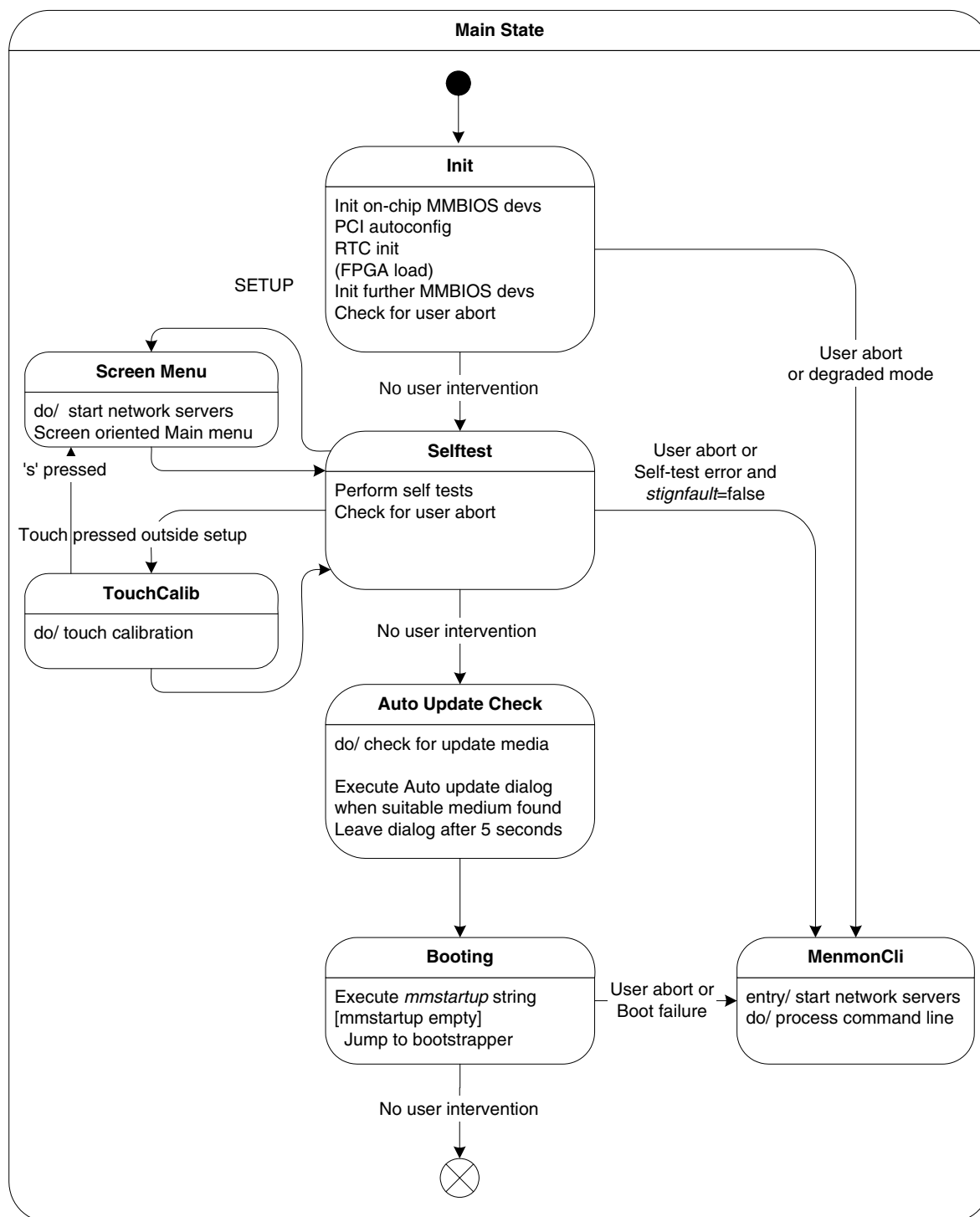


Figure 6. MENMON — State Diagram, Main State

4.2 Interacting with MENMON

To interact with MENMON, you can use the following consoles:

- COM1/COM2 serial interfaces
- Touch panel / TFT interface (if present)
- Telnet via network connection
- HTTP monpage via network connection

The default setting of the COM ports is 9600 baud, 8 data bits, no parity, and one stop bit.

4.2.1 Entering the Setup Menu/Command Line

During normal boot, you can abort the booting process in different ways during the self-test, depending on your console:

- With a touch panel press the "Setup" button to enter the Setup Menu.
- With a text console press the "s" key to enter the Setup Menu.
- With a text console press "ESC" to enter the command line.

By default, the self-test is not left until 3 seconds have elapsed (measured from the beginning of the self-test), even if the actual test has finished earlier, to give the user a chance to abort booting and enter the Setup Menu.

4.3 Configuring MENMON for Automatic Boot

You can configure how MENMON boots the operating system either through the Setup Menu or through the command line.

In the Basic Setup Menu you can select the boot sequence for the bootable devices on EM4N. The selected sequence is stored in system parameter *mmstartup* as a string of MENMON commands. For example, if the user selects: "Int. CF, Ether, (None)", the *mmstartup* string will be set to "DBOOT 0; NBOOT TFTP".

You can view and modify this string directly, using the Expert Setup Menu, option *Startup string*, or through the command-line command *EE-MMSTARTUP*.



(See also [MENMON 2nd Edition User Manual](#) for further details.)

4.4 Calibrating the Touch Screen

You can enter the touch-panel calibration function through the Setup Main Menu.

This function is also entered automatically during the self-test, if you hit the touch screen at any position outside the "Setup" button. You may have missed the "Setup" button because the touch panel was incorrectly calibrated.

Follow the instructions on the screen to complete calibration.



(See also [MENMON 2nd Edition User Manual](#) for further details.)

4.5 Updating Boot Flash and CompactFlash

Primary MENMON is hardware protected.

Please note that MENMON 3.x as primary MENMON cannot start older versions as secondary (because older versions do not appear to have a valid header).

4.5.1 Update via the Serial Console using *SERDL*

You can use command *SERDL* to update program data using the serial console.

The following table shows the EM4N locations:

Table 14. MENMON — Program Update Files and Locations

File Name Extension	Typical File Name	Password for <i>SERDL</i>	Location
.SMM	MENMON_EM04.SMM	MENMON	Secondary MENMON
.FP0	EM04N11IC002A1.FP0	FPGA0	FPGA0 code
.FP1	EM04N11IC002A1.FP1	FPGA1	FPGA1 code (backup)
.Fxxx	MYFILE.F000	-	Starting at sector xxx in boot Flash
.Exx	MYFILE.E00	-	Starting at byte xx in EEPROM
.Cxx	DSKIMG.C00	DISK	Starting at sector xx in internal CompactFlash
.Bxx	DSKIMG.B00	DISK	Starting at sector xx in external CompactFlash

4.5.2 Update from Network using *NDL*

You can use the network download command *NDL* to download the update files from a TFTP server in network. The file name extensions, locations and passwords are the same as for the *SERDL* command.

4.5.3 Update via Program Update Menu

The following Program Update Menu is implemented in the EM4N MENMON:

Program Update Menu

```
>Copy external CF -> internal CF (1:1)

Copy external CF:IMAGE.C00 -> internal CF

Copy external CF:IMAGE.FP0 -> boot flash FPGA code

Copy external CF:IMAGE.FP1 -> boot flash fallback FPGA code

Copy external CF:IMAGE.SMM -> boot flash sec. MENMON

Copy internal CF -> external CF (1:1)
```

4.5.4 Automatic Update Check

On EM4N MENMON's automatic update check looks for some special files on the external CompactFlash card.

The files that are searched for are:

- Name stored in system parameter *bf* or *bootfile*, or—if this is empty—*BOOTFILE*¹—If this file is found, it is assumed that the entire external CompactFlash is supposed to be copied to internal Flash.
- *IMAGE.C00*—If this file is found, it is assumed that *IMAGE.C00* contains an image of a CompactFlash card.

To allow MENMON to locate these files, they must be in the root directory of a DOS FS. This works on unpartitioned media or on drives with one partition.

MENMON does not automatically start the copying process. Depending on the type of file found, it presents different menus to the user:

In case *BOOTFILE* was found:

```
Detected an update capable external CompactFlash

>Ignore, continue boot

Copy external CF -> internal CF (1:1)

Boot from external CompactFlash
```

In case *IMAGE.C00* was found:

```
Detected an update capable external CompactFlash

>Ignore, continue boot

Copy external CF:IMAGE.C00 -> internal CF
```



The copying process is then performed in the same way as a standard sector-by-sector media copy program update (see [MENMON 2nd Edition User Manual](#)).

If there is no user input for 5 seconds after the menu appears, booting continues.

¹ MENMON versions < 3.4 only search for *BOOTFILE*.

4.6 Diagnostic Tests

Note: MENMON may include further tests for COM or other interfaces depending on the EM4N functionality and carrier board. The standard carrier board tests are:

- TOUCH
- IDE
- COM_CB

4.6.1 Ethernet

Table 15. MENMON — Diagnostic Tests: Ethernet

Test Name	Description	Availability
<i>ETHER0</i>	Ethernet 0 (LAN1) internal loopback test Groups: POST AUTO ENDLESS	Always
<i>ETHER1</i>	Ethernet 1 (LAN2) internal loopback test Groups: POST AUTO ENDLESS	MENMON BIOS device "ETHER1" present
<i>ETHER0_X</i>	Ethernet 0 (LAN1) external loopback test Groups: POST AUTO ENDLESS	Always
<i>ETHER1_X</i>	Ethernet 1 (LAN2) external loopback test Groups: POST AUTO ENDLESS	MENMON BIOS device "ETHER1" present

4.6.1.1 Ethernet Internal Loopback Test

The test

- configures the network interface for loopback mode (on PHY)
- verifies that the interface's ROM has a good checksum
- verifies that the MAC address is valid (not 0xFFFFFFFF...)
- sends 10 frames with 0x400 bytes payload each
- verifies that frames are correctly received on the same interface.

If the network interface to test is the currently activated interface for the MENMON network stack, the interface is detached from the network stack during test and reactivated after test.

Checks:

- Connection between CPU and LAN controller
- Connection between LAN controller and PHY

Does not check:

- Connection between PHY and physical connector
- Interrupt line
- All LAN speeds

4.6.1.2 Ethernet External Loopback Test

This test is the same as the Ethernet Internal Loopback Test, but requires an external loopback connector. Before sending frames, the link state is monitored. If it is not ok within 2 seconds, the test fails.

Checks:

- Connection between CPU and LAN controller
- Connection between LAN controller and PHY
- Connection between PHY and physical connector

Does not check:

- Interrupt line
- All LAN speeds

4.6.2 SDRAM and SRAM

Table 16. MENMON — Diagnostic Tests: SDRAM and SRAM

Test Name	Description	Availability
<i>SDRAM</i>	Quick SDRAM connection test Groups: POST AUTO	Always
<i>SDRAM_X</i>	Full SDRAM test Groups: NONAUTO ENDLESS	Always
<i>SRAM</i>	Quick SRAM test Groups: POST AUTO	SRAM IP core was found in main FPGA and carrier board is known to have SRAM
<i>SRAM_X</i>	Full SRAM test Groups: NONAUTO ENDLESS	

4.6.2.1 Quick RAM Test

This quick test checks most of the connections to the RAM chips but does not test all RAM cells. It executes very quickly (within milliseconds).

This test is non-destructive (saves/restores original RAM content).

Checks:

- All address lines
- All data lines
- Byte enable signals
- Indirectly, checks clock and other control signals

Does not check:

- SDRAM cells
- SO-DIMM SPD EEPROM
- Burst mode

4.6.2.2 Extended RAM Test

This full-featured memory test allows to test all RAM cells. Depending on the size of the SO-DIMM, this test can take up to one minute.

It tests 8-, 16- or 32-bit access, each with random pattern, and single and burst access.

On each pass, this test first fills the entire memory (starting with the lowest address) with the selected pattern, using the selected access mode, and then verifies the entire block.

This test is destructive.

Checks:

- All address lines
- All data lines
- All control signals
- All SDRAM cells

Does not check:

- SO-DIMM SPD EEPROM

4.6.3 EEPROM

Table 17. MENMON — Diagnostic Tests: EEPROM

Test Name	Description	Availability
EEPROM	I2C access/Magic nibble check Groups: POST AUTO ENDLESS	Always

This test reads the first EEPROM cell over SMB and checks if bits 3..0 of this cell contain the magic nibble 0xD.

4.6.4 IDE/CompactFlash

Table 18. MENMON — Diagnostic Tests: IDE/CompactFlash

Test Name	Description	Availability
<i>IDE</i>	External IDE master access / sector 0 access Groups: NONAUTO ENDLESS	MENMON BIOS device 1/0 present and carrier board is known to have external IDE
<i>INT_CF</i>	Internal CompactFlash access / sector 0 access Groups: POST AUTO ENDLESS	MENMON BIOS device 0/0 present
<i>EXT_CF</i>	External CompactFlash access / sector 0 access Groups: NONAUTO ENDLESS	MENMON BIOS device 1/0 present and carrier board is known to have external CompactFlash

The test first performs an ATA register test, then reads sector 0 from the Flash disk without verifying the content of the sector.

Checks:

- Most ATA control lines
- Basic ATA transfer

Does not check:

- ATA signals IRQ, DAK, DRQ
- Partition table or file system on disk

4.6.5 COM1/COM2

Table 19. MENMON — Diagnostic Tests: COM1/COM2

Test Name	Description	Availability
COM1	External loopback test RxD/TxD Groups: NONAUTO ENDLESS Note: Test will be SKIPPED when COM1 is currently used as a console	Always
COM2	External loopback test RxD/TxD Groups: NONAUTO ENDLESS	When no carrier board specific "COM2" diagnostic test is registered

This test requires an external test adapter connecting:

- TXD and RXD
To test TXD/RXD, a test string is sent through the UART.

To test TxD/RxD, a test string is sent through the UART.

To test handshake lines, the lines are toggled and it is checked whether input lines follow.

4.6.6 Touch

Table 20. MENMON — Diagnostic Tests: Touch

Test Name	Description	Availability
TOUCH	Touch controller communication test Groups: POST AUTO ENDLESS	Carrier board is known to have a touch controller

This test tries to communicate over the SPI bus with the touch controller on the carrier board by sending an Identify command to the controller.

Checks:

- SPI connection to touch controller

Does not check:

- Connection between touch controller and touch panel

4.6.7 Primary/Secondary MENMON

Table 21. MENMON — Diagnostic Tests: Primary/Secondary MENMON

Test Name	Description	Availability
CHECKSUM PMM	Checksum Primary MENMON Groups: POST AUTO	Always
CHECKSUM SMM	Checksum Secondary MENMON Groups: POST AUTO	Always

4.6.8 RTC

Table 22. MENMON — Diagnostic Tests: RTC

Test Name	Description	Availability
<i>RTC</i>	Quick presence test of RTC Groups: POST AUTO	Always
<i>RTC_X</i>	Extended test of RTC Groups: NONAUTO ENDLESS	Always

4.6.8.1 RTC Test

This is a quick presence test of the real-time clock (RTC) and is executed on POST.

Checks:

- Presence of RTC (I2C access)

Does not check:

- If RTC is running
- RTC backup voltage

4.6.8.2 Extended RTC Test

Checks:

- Presence (e.g. I2C access)
- RTC is running

Does not check:

- RTC backup voltage

4.7 MENMON Configuration and Organization

4.7.1 Consoles

You can select the active consoles by means of system parameters *con0..con3*:

Table 23. MENMON — System Parameters for Console Selection

Parameter (alias)	Description	Default	User Access
<i>con0..con3</i>	CLUN of console 0..3. CLUN=0x00: disable CLUN=0xFF: Autoselect next available console <i>con0</i> is implicitly the debug console	<i>con0</i> : 08 (COM1) <i>con1</i> : 0A (Touch) <i>con2</i> : 00 (none) <i>con3</i> : 00 (none)	Read/write
<i>gcon</i>	CLUN of graphics device to display boot logo CLUN=0x00: disable CLUN=0xFF: Autoselect first available graphics console	0xFF (AUTO)	Read/write

Some EM4N models have no COM1. MENMON checks at every system start if it runs on such a model and temporarily sets the *con3* parameter to 0x09 (COM2). The console will then appear on COM1 and 2.

4.7.2 Video Modes

None of the included drivers allows to change the video mode.

4.7.3 Abort Pin

Since EM4N has no real "abort" button, it is simulated by connecting pin 1 to pin 2 on the debug connector (TDI pin of debugger with GND).

If the abort pin is detected asserted, the secondary MENMON is not invoked and MENMON uses default parameters (such as baud rate, console port). This is useful if a secondary MENMON has been programmed that does not work or if you have misconfigured a system parameter.

Note that when a JTAG debugger is connected, the abort pin is always read as active.

4.7.4 MENMON Memory Map

4.7.4.1 MENMON Memory Address Mapping

Table 24. MENMON — Address Map (Full-featured mode)

Address Space	Size	Description
0x 01F0 0000 .. 01F7 FFFF	512KB	Text + Reloc
0x 01F8 0000 .. 01F8 FFFF	64KB	Stack
0x 01F9 0000 .. 01F9 FFFF	64KB	Stack for user programs
0x 01FA 0000 .. 01FF FFFF	384KB	Heap1
0x 01D0 0000 .. 01EF FFFF	2MB	Heap2

4.7.4.2 Boot Flash Memory Map

Table 25. MENMON — Boot Flash Memory Map (2MB)

Address Space	Description
0x FFE0 0000 .. FE07 FFFF	Up to 512KB initial FPGA code
0x FFE8 0000 .. FFEF FFFF	Optional: up to 512KB fallback FPGA code. Available to user if fallback FPGA code not used
0x FFF0 0000 .. FF7F FFFF	Primary MENMON
0x FFF8 0000 .. FFFF BFFF	Secondary MENMON
0x FFFF C000 .. FFFF FFFF	16KB for system parameters (used when system parameter <i>useflpar</i> is 1)

4.7.5 MENMON BIOS Logical Units

The following table shows fixed assigned CLUNs. All other CLUNs are used dynamically.

Table 26. MENMON — Controller Logical Units (CLUNs)

CLUN	MENMON BIOS Name	Description
0x00	IDE0	Primary IDE = internal CompactFlash
0x01	IDE1	Secondary IDE (e.g. external CompactFlash)
0x02	ETHER0	Ethernet LAN1
0x03	ETHER1	Ethernet LAN2
0x08	COM1	DUART8245 channel #0 (COM1)
0x09	COM2 COM2H	DUART8245 channel #1 (COM2) Name is COM2H if supported by helper unit
0x0A	TOUCH	Touch console (if 16Z031_SPI found in on-board FPGA and can communicate with touch controller)
0x0B	COM10	UART COM10 of on-board FPGA UART
0x0C	COM11	UART COM11 of on-board FPGA UART
0x0D	COM12	UART COM12 of on-board FPGA UART
0x0E	COM13	UART COM13 of on-board FPGA UART
0x10	CAN_TEST	First instance of CAN_TEST unit in on-board FPGA
0x20		All other devices dynamically detected on PCI or FPGA devices
0x40		Telnet console
0x41		HTTP monitor console

Table 27. MENMON — Device Logical Units (DLUNs)

CLUN/DLUN	MENMON BIOS Name	Description
0x00/0x00	Int. CF	Internal CompactFlash
0x01/0x00	Ext. CF	If 16Z023_IDE present, external CompactFlash
0x01/0x00 0x01/0x01	IDE1-M IDE1-S	If 16Z023_IDE_NHS present, non-hot-swappable IDE

4.7.6 System Parameters

System parameters are parameters stored in EEPROM. Some parameters are automatically detected by MENMON (such as CPU type and frequency). The parameters can be modified through the *EE-xxx* command via the command line.

4.7.6.1 Backward Compatibility of Parameters

MENMON 3.x for EM4N is backward-compatible to older MENMON versions:

- Existing EEPROM sections are kept unchanged.
- A new EEPROM section (*menmx_parms*) has been added at offset 0x190..0x19F, which mainly stores the network and console configuration.
- A new section in the last sector of boot Flash has been added. It is only used when parameter *useflpar* has been set to 1.

4.7.6.2 Physical Storage of Parameters in Ethernet SROMs

On EM4N, LAN1 and 2 provide their own SROM, the parameters *nspeedX* and *nmacX* appear as system parameters.

4.7.6.3 EM4N System Parameters

Note: Parameters marked by "Yes" in section "Parameter String" are part of the MENMON parameter string.

 A green background marks parameters different to the global specification.

 A blue background marks parameters different/new to older MENMON versions.

Table 28. MENMON — EM4N System Parameters — Autodetected Parameters

Parameter (alias)	Description	Standard Default	Parameter String	User Access
<i>clun</i>	MENMON controller unit number that MENMON used as the boot device (hexadecimal)		Yes	Read-only
<i>cons</i>	Selected console. Board specific, for backward compatibility. Setup according to device name that is used for <i>con0</i>		Yes	Read-only
<i>cpu</i>	CPU type as ASCII string (e.g. "MPC8245")		Yes	Read-only
<i>cpuclockhz</i>	CPU core clock frequency (decimal, Hz)		Yes	Read-only
<i>dlun</i>	MENMON device unit number that MENMON used as the boot device (hexadecimal)		Yes	Read-only
<i>flash[0..n]</i>	Boot Flash size (decimal, kilobytes)		Yes	Read-only
<i>mem0</i>	RAM size (decimal, kilobytes)		Yes	Read-only
<i>mem1..n</i>	Board-specific size of additional memory (carrier board SRAM)		Yes	Read-only

Parameter (alias)	Description	Standard Default	Parameter String	User Access
<i>memclkhz</i>	Memory clock frequency (decimal, Hz)		Yes	Read-only
<i>mm</i>	Info whether primary or secondary MENMON has been used for booting, either "smm" or "pmm"		Yes	Read-only
<i>mmst</i>	Status of diagnostic tests, as a string		Yes	Read-only
<i>pciclkhz</i>	PCI bus clock frequency (decimal, Hz)		Yes	Read-only
<i>rststat</i>	Reset status code as a string, see Chapter 4.7.6.4 Parameter rststat (Reset Cause) on page 66		Yes	Read-only

Table 29. MENMON — EM4N System Parameters — Production Data

Parameter (alias) ¹	Description	Standard Default	Parameter String	User Access
<i>brd</i>	Board name	-	Yes	Read-only
<i>brdmod</i>	Board model "mm"	-	Yes	Read-only
<i>brdrev</i>	Board revision "xx.yy.zz"	-	Yes	Read-only
<i>prodat</i>	Board production date MM/DD/YYYY	-	Yes	Read-only
<i>repdat</i>	Board last repair date MM/DD/YYYY	-	Yes	Read-only
<i>sernbr</i>	Board serial number	-	Yes	Read-only

¹ Parameters for production data of carrier boards will use prefixed parameter names, e.g. *c-brd*.

Table 30. MENMON — EM4N System Parameters — Persistent Parameters

Parameter (alias)	Description	Standard Default	Parameter String	User Access
<i>bsadr (bs)</i>	Bootstrapper address. Used when BO command was called without arguments. (hexadecimal, 32 bits)	0	No	Read/write
<i>cbr (baud)</i>	Baudrate of all UART consoles (dec)	9600	Yes	Read/write
<i>com3se</i>	Set COM3 to single ended mode (default=RS422, FDX)	0		Read/write
<i>con0..conN</i>	CLUN of console 0..n. (hex) (see Chapter 4.7.1 Consoles on page 59)	Depends on implementation	No	Read/write
<i>ecl</i>	CLUN of attached network interface (hex)	0xFF	No	Read/write
<i>gcon</i>	CLUN of graphics screen (hex) (see Chapter 4.7.1 Consoles on page 59)	0xFF = auto	No	Read/write
<i>hdp</i>	HTTP server TCP port (decimal)	-1	No	Read/write
<i>kerpar</i>	Linux Kernel Parameters (399 chars max)	Empty string	No	Read/write
<i>ldlogodis</i>	Disable load of boot logo (bool)	0	No	Read/write

Parameter (alias)	Description	Standard Default	Parameter String	User Access
<i>mmstartup (startup)</i>	Start-up string (511 chars max)	Empty string	No	Read/write
<i>nmacX</i>	MAC address of Ethernet interface x (0..n). Format e.g. "00112233445566". Stored in dedicated ROM of Ethernet device.	0xFFFFFFFFFFFF	Yes	Read/write
<i>nobanner</i>	Disable ASCII banner on start-up	0	No	Read/write
<i>nspeedX</i>	Speed setting for Ethernet interface x (0..n). Stored in dedicated ROM of Ethernet device. Possible values: <i>AUTO</i> , <i>10HD</i> , <i>10FD</i> , <i>100HD</i> , <i>100FD</i>	AUTO	Yes	Read/write
<i>selftest</i>	For backward compatibility. All self-test flags (hex)	0		Read/write
<i>stdis</i>	Disable POST (bool)	0	No	Read/write
<i>stdis_int_cf</i>	Disable internal CompactFlash test	0	No	Read/write
<i>stdis_prtctrl</i>	Disable printer controller test	0	No	Read/write
<i>stdis_sram</i>	Disable SRAM test	0	No	Read/write
<i>stdis_touch</i>	Disable touch test	0	No	Read/write
<i>stignfault</i>	Ignore POST failure, continue boot (bool)	0	No	Read/write
<i>stwait</i>	Time in 1/10 seconds to stay at least in SELFTEST state (decimal) 0 = Continue as soon as POST has finished	30	No	Read/write
<i>tdp</i>	Telnet server TCP port (decimal)	-1	No	Read/write
<i>tries</i>	Number of network tries	20	No	Read/write
<i>tto</i>	Minimum timeout between network retries (decimal, in seconds)	0	No	Read/write
<i>u00..u15</i>	User parameters (hex, 16 bits)	0x0000	No	Read/write
<i>updcdis</i>	Disable auto update check (bool)	0	No	Read/write
<i>useflpar</i>	Store "kerpar" and "startup" parameters in boot Flash rather than in EEPROM. Parameter is used on boards where backward compatibility must be preserved (bool)	0	No	Read/write

Parameter (alias)	Description	Standard Default	Parameter String	User Access
<i>vmode</i>	Vesa Video Mode for graphics console (hex) (see Chapter 4.7.2 Video Modes on page 59)	0x0101	No	Read/write
<i>wdt</i>	Time after which watchdog timer shall reset the system after MENMON has passed control to operating system (decimal, in 1/10 s) If 0, MENMON disables the watchdog timer before starting the operating system.	0 (disabled)	No	Read/write

Table 31. MENMON — EM4N System Parameters — VxWorks Bootline Parameters

Parameter (alias)	Description	Standard Default	Parameter String	User Access
<i>bf (bootfile)</i>	Boot file name (127 chars max)	Empty string	No	Read/write
<i>bootdev</i>	VxWorks boot device name	Empty string	No	Read/write
<i>e (netip)</i>	IP address, subnet mask, e.g. 192.1.1.28:ffffff00	Empty string	No	Read/write
<i>g (netgw)</i>	IP address of default gateway	Empty string	No	Read/write
<i>h (nethost)</i>	Host IP address (used when booting over <i>NBOOT TFTP</i>)	Empty string	No	Read/write
<i>hostname</i>	VxWorks name of boot host	Empty string	No	Read/write
<i>netaddr</i>	Access the IP address part of <i>netip</i> parameter		No	Read/write
<i>netsm</i>	Access the subnet mask part of <i>netip</i> parameter		No	Read/write
<i>procnum</i>	VxWorks processor number (decimal)	0	No	Read/write
<i>s</i>	VxWorks start-up script	Empty string	No	Read/write
<i>tn (netname)</i>	Host name of this machine	Empty string	No	Read/write
<i>unitnum</i>	VxWorks boot device unit number (decimal)	0	No	Read/write

Table 32. MENMON — EM4N System Parameters — Carrier-Board Specific Parameters

Parameter (alias)	Description	Standard Default	Parameter String	User Access
<i>c-tcal</i>	Touch calibration parameters	0,0,0,0	Yes	Read/write

4.7.6.4 Parameter *rststat* (Reset Cause)

On EM4N, the following *rststat* values are possible:

When MENMON starts up, it determines the reset cause and sets system parameter *rststat* accordingly:

Table 33. MENMON — Reset Causes through System Parameter *rststat*

<i>rststat</i> Value	Description
<i>hrst</i>	Board was reset due to activation of HRESET line
<i>pwon</i>	Power On
<i>rbut</i>	Board was reset by an external reset pin (e.g. reset button)
<i>swrst:nn</i> <i>swrst</i>	Board was reset by software (by means of the board's reset controller). <i>nn</i> is the hexadecimal value of an additional register that can be set through software (e.g. system unit GPMEM) register. If <i>:nn</i> is omitted, the system unit does not support an additional register (e.g. on 16Z000). The following values are defined for <i>nn</i>: 00 = No special reason 80 = OS panic (general)
<i>wdog</i>	Board was reset by watchdog timer unit

4.8 MENMON Commands

The following table gives all MENMON commands that can be entered on the EM4N MENMON prompt.

 A green background marks commands different to the global specification.

Table 34. MENMON — Command Reference

Command	Description
.[<reg>] [<val>]	Display/modify registers in debugger model
ARP	Dump network stack ARP table
AS <addr> [<cnt>]	Assemble memory
B[DC#] [<addr>]	Set/display/clear breakpoints
BIOS_DBG <mask> [net] cons <clun>	Set MENMON BIOS or network debug level, set debug console
BO [<addr>] [<opts>]	Call OS bootstrapper
BOOTP [<opts>]	Obtain IP config via BOOTP
C[BWLLNAX#] <addr> [<val> ...]	Change memory
CHAM-LOAD [<addr>]	Load FPGA
CHAM [<clun>]	Dump FPGA Chameleon table
CONS	Show active consoles
CONS-ACT <clun1> [<clun2>] ...	Test console configuration
D [<addr>] [<cnt>]	Dump memory
DBOOT [<clun>] [<dlun>] [<opts>]	Boot from disk
DCACHE OFFION	Enable/disable data cache
DI [<addr>] [<cnt>]	Disassemble memory
DIAG [<which>] [VTF]	Run diagnostic tests
DSKWR <args>	Write blocks to RAW disk
DSKRD <args>	Read blocks from RAW disk
EER[-xxx] [<arg>]	Raw serial EEPROM commands
EE[-xxx] [<arg>]	Persistent system parameter commands
ERASE <D> [<O>] [<S>]	Erase Flash sectors
ESMCB-xxx	ESM carrier commands
FI <from> <to> <val>	Fill memory (byte)
GO [<addr>]	Jump to user program
H HELP	Print help
I [<D>]	List board information
ICACHE OFFION	Enable/disable instruction cache
IOI	Scan for BIOS devices
LATCH [<val>]	MEN internal test command

Command	Description
LOGO	Display MENMON start-up screen
LS <clun> <dlun> [<opts>]	List files/partitions on device
MC <addr1> <addr2> <cnt>	Compare memory
MII <clun> [<reg>] [<val>]	Ethernet MII register command
MO <from> <to> <cnt>	Move (copy) memory
MS <from> <to> <val>	Search pattern in memory
MT [<opts>] <start> <end> [<runs>]	Memory test
NBOOT [<opts>]	Boot from Network
NDL [<opts>]	Update Flash from network
NETSTAT	Show current state of networking parameters
PCI-VPD[-] <devNo> [<busNo>] [<capId>]	PCI Vital Product Data dump
PCIC <dev> <addr> [<bus>] [<func>]	PCI config register change
PCID[+] <dev> [<bus>] [<func>]	PCI config register dump
PCI	PCI probe
PCIR	List PCI resources
PFLASH <D> <O> <S> [<A>]	Program Flash
PGM-XXX <args>	Media copy tool
PING <host> [<opts>]	Network connectivity test
RST	Cause an instant system reset
RTC[-xxx] [<arg>]	Real time clock commands
S [<addr>]	Single step user program
SERDL [<passwd>]	Update Flash using YModem protocol
SETUP	Open Setup Menu

5 Organization of the Board

To install software on the board or to develop low-level software it is essential to be familiar with the board's address and interrupt organization.

5.1 Memory Mappings

The memory mapping of the board complies with the PowerPC CHRP (Common Hardware Reference Platform) Specification. The integrated host-to-PCI bridge is set to map B to support this mapping.

Table 35. Memory Map — Processor View

CPU Address Range	Size	Description
0x 0000 0000 .. 3FFF FFFF	1GB	SDRAM (upper limit depends on SDRAM size)
0x 7000 0000 .. 7FFF FFFF	256MB	Extended "ROM" space
0x 8000 0000 .. FCFF FFFF	2GB-48MB	PCI Memory Space
0x FD00 0000 .. FDFF FFFF	16MB	PCI ISA Memory Space
0x FE00 0000 .. FE00 FFFF	64KB	PCI ISA I/O Space
0x FE80 0000 .. FEBF FFFF	4MB	PCI I/O Space (not used)
0x FEC0 0000 .. FEEF FFFF	2MB	PCI Config Addr. Registers
0x FEFO 0000 .. FEFF FFFF	1MB	PCI IACK Space (not used)
0x FF00 0000 .. FFFF FFFF	Max. 16MB	Boot Flash (8-bit)

Table 36. Address Mapping for PCI

Address Range	Description
<i>PCI Memory Space (addresses as seen on PCI bus)</i>	
0x 8000 0000 .. 81FF FFFF	FPGA: up to 32MB for prefetchable BARs
0x 8200 0000 .. 8FFF FFFF	Available for PCI auto configuration, prefetchable BARs
0x 9000 0000 .. 91FF FFFF	FPGA up to 32MB non-prefetchable BARs
0x 9200 0000 .. FBFF FFFF	Available for PCI auto configuration, non-prefetchable BARs
0x FC00 0000 .. FC0F FFFF	MPC8245 embedded utility block (internal devices)

Table 37. BATS set up by MENMON¹

BAT	Address Range	Attributes	Description
IBAT0 DBAT0	0x FF00 0000 .. FFFF FFFF	Cache I/O	Flash
IBAT1 DBAT1	0x 0000 0000 .. 0FFF FFFF	Cache I/O	DRAM Changed after DRAM init to Cache
DBAT2	0x 9000 0000 .. 9FFF FFFF 0x 7000 0000 .. 7FFF FFFF	I/O	PCI Memory space non-prefetchable RCS2 latch DBAT2 is swapped by MENMON as needed
DBAT3	0x D000 0000 .. DFFF FFFF 0x xxxx xxxx .. xxxx xxxx	Cache xxxx	L1 cache as RAM Used for BAT swapping (in MENMON full mode)

5.2 Interrupt Handling

Interrupt handling is done by the FPGA and the MPC8245's EPIC, which is used in Serial Interrupt Mode. Every interrupt, i.e. the interrupt of every module in the FPGA, can be enabled or disabled. In addition the MPC8245 can mask and prioritize all 16 interrupts in EPIC.

For a more detailed description see [Chapter 3.2.1.1 Interrupt Controller on page 38](#) in the FPGA section of this manual.

5.3 SMB Devices

Table 38. SMB Devices

Address	Function
0xA0	SPD of SO-DIMM
0xA8 / 0xAA	CPU EEPROM (512 bytes)
0xAC / 0xAE	Carrier board EEPROM (512 bytes)
0xD0	RTC M41T56

¹ Unless otherwise stated, all BATS are initialized with W I M !G.

5.4 PCI Devices on Bus 0

Table 39. *PCI Devices on Bus 0*

Device Number	Vendor ID	Device ID	Function	Interrupt
0x00	0x1057	0x0006	PPC/PCI bridge in MPC8245	-
0x1D	0x1172	0x4D45	FPGA	
0x1A	0x8086	0x1209	Ethernet LAN1	INTC
0x1B	0x8068	0x1208	Ethernet LAN2	INTD
0x14			PCI-104 slot 1	INTA
0x15			PCI-104 slot 2	INTB
0x16			PCI-104 slot 3	INTC
0x17			PCI-104 slot 4	INTD

6 Appendix



6.1 Literature and Web Resources

6.1.1 PowerPC

- MPC8245:
MPC8245 Integrated Processor Reference Manual
MPC8245UM; 2005; Freescale Semiconductor, Inc.
www.freescale.com

6.1.2 EIDE

- EIDE:
Information Technology - AT Attachment-3 Interface (ATA-3), Revision 6,
working draft; 1995; Accredited Standards Committee X3T10

6.1.3 Ethernet

- Ethernet in general:
 - The Ethernet, A Local Area Network, Data Link Layer and Physical Layer Specifications, Version 2.0; 1982; Digital Equipment Corporation, Intel Corp., Xerox Corp.
 - ANSI/IEEE 802.3-1996, Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications; 1996; IEEE
www.ieee.org
- www.ethermanage.com/ethernet/
links to documents describing Ethernet, components, media, the Auto-Negotiation system, multi-segment configuration guidelines, and information on the Ethernet Configuration Guidelines book
- www.iol.unh.edu/training/ethernet.html
collection of links to Ethernet information, including tutorials, FAQs, and guides
- ckp.made-it.com/ieee8023.html
Connectivity Knowledge Platform at Made IT technology information service, with lots of general information on Ethernet

6.1.4 PCI-104

- PCI-104:
PCI-104 Specification; PC/104 Embedded Consortium
www.pc104.org

You can request the circuit diagrams for the current revision of the product described in this manual by completely filling out and signing the following non-disclosure agreement.

Please send the agreement to MEN by mail. We will send you the circuit diagrams along with a copy of the completely signed agreement by return mail.

MEN reserves the right to refuse sending of confidential information for any reason that MEN may consider substantial.



Non-Disclosure Agreement

for Circuit Diagrams provided by MEN Mikro Elektronik GmbH

between

MEN Mikro Elektronik GmbH
Neuwieder Straße 5-7
D-90411 Nürnberg

("MEN")

and

("Recipient")

We confirm the following Agreement:

MEN

Date: _____

Name: _____

Function: _____

Recipient

Date: _____

Name: _____

Function: _____

Signature:

Signature:

The following Agreement is valid as of the date of the MEN signature.

MEN Mikro Elektronik GmbH

Neuwieder Straße 5-7
90411 Nürnberg
Deutschland

Tel. +49-911-99 33 5-0
Fax +49-911-99 33 5-901

E-Mail info@men.de
www.men.de

1 Subject

The subject of this Agreement is to protect all information contained in the circuit diagrams of the following product:

Article Number: _____ [filled out by recipient]

MEN provides the recipient with the circuit diagrams requested through this Agreement only for information.

2 Responsibilities of MEN

Information in the circuit diagrams has been carefully checked and is believed to be accurate as of the date of release; however, no responsibility is assumed for inaccuracies. MEN will not be liable for any consequential or incidental damages arising from reliance on the accuracy of the circuit diagrams. The information contained therein is subject to change without notice.

3 Responsibilities of Recipient

The recipient, obtaining confidential information from MEN because of this Agreement, is obliged to protect this information.

The recipient will not pass on the circuit diagrams or parts thereof to third parties, neither to individuals nor to companies or other organizations, without the written permission by MEN. The circuit diagrams may only be passed to employees who need to know their content. The recipient protects the confidential information obtained through the circuit diagrams in the same way as he protects his own confidential information of the same kind.

4 Violation of Agreement

The recipient is liable for any damage arising from violation of one or several sections of this Agreement. MEN has a right to claim damages amounting to the damage caused, at least to €100,000.

5 Other Agreements

MEN reserves the right to pass on its circuit diagrams to other business relations to the extent permitted by the Agreement.

Neither MEN nor the recipient acquire licenses for the right of intellectual possession of the other party because of this Agreement.

This Agreement does not result in any obligation of the parties to purchase services or products from the other party.

6 Validity of Agreement

The period after which MEN agrees not to assert claims against the recipient with respect to the confidential information disclosed under this Agreement shall be _____ months [filled out by MEN]. (Not less than twenty-four (24) nor more than sixty (60) months.)

7 General

If any provision of this Agreement is held to be invalid, such decision shall not affect the validity of the remaining provisions and such provision shall be reformed to and only to the extent necessary to make it effective and legal.

This Agreement is only effective if signed by both parties.

Amendments to this Agreement can be adopted only in writing. There are no supplementary oral agreements.

This Agreement shall be governed by German Law.

The court of jurisdiction shall be Nuremberg.



MEN Mikro Elektronik GmbH

Neuwieder Straße 5-7
90411 Nürnberg
Deutschland

Tel. +49-911-99 33 5-0
Fax +49-911-99 33 5-901

E-Mail info@men.de
www.men.de